E. Floegel

Anwender-programme

für Apple II c® und Apple II e®

Grafikprogramme · Dateiverwaltungen ·
Datenstrukturen · Geschäftsprogramme

HOFACKER

ISBN 3-88963-224-6

Es kann keine Gewähr dafür übernommen werden, daß die in diesem Buche verwendeten Angaben, Schaltungen, Warenbezeichnungen und Warenzeichen, sowie Programmlistings frei von Schutzrechten Dritter sind. Alle Angaben werden nur für Amateurzwecke mitgeteilt. Alle Daten und Vergleichsangaben sind als unverbindliche Hinweise zu verstehen. Sie geben auch keinen Aufschluß über eventuelle Verfügbarkeit oder Liefermöglichkeit. In jedem Falle sind die Unterlagen der Hersteller zur Information heranzuziehen.

Nachdruck und öffentliche Wiedergabe, besonders die Übersetzung in andere Sprachen verboten. Programmlistings dürfen weiterhin nicht in irgendeiner Form vervielfältigt oder verbreitet werden. Alle Programmlistings sind Copyright der Fa. Ing. W. Hofacker GmbH. Verboten ist weiterhin die öffentliche Vorführung und Benutzung dieser Programme in Seminaren und Ausstellungen. Irrtum, sowie alle Rechte vorbehalten.

COPYRIGHT by Ing. W. HOFACKER ©1984
Tegernseer Str. 18, 8150 Holzkirchen

1. Auflage 1984

Gedruckt in der Bundesrepublik Deutschland – Printed in West-Germany – Imprime'en RFA.

E. Floegel

Anwenderprogramme
für Apple II c® und Apple II e®

E. Floegel

Anwenderprogramme
für Apple II c® und Apple II e®

Grafikprogramme ✳ Dateiverwaltungen ✳
Datenstrukturen ✳ Geschäftsprogramme

Inhaltsverzeichnis

1. Vorwort und Einführung 1
2. Aufteilen von Zeichenketten 3
3. Verkettete Listen 7
   3.1 Lineare Listen 8
   3.2 Doppelt verkettete Listen 11
4. Hashing 13
5. Registrierkasse 19
6. Notizbuch 29
7. Stichwort-Kartei 35
8. AVL-Bäume 45
9. Zeichnen eines Balkendiagramms 59
10. Zeichnen eines Tortendiagramms 65
11. Darstellung ebener Funktionen 69
12. Darstellung räumlicher Funktionen 73
13. Computer-Grafiken 79
   13.1 Polygon-Graphik 79
   13.2 Irrgarten 82
   13.3 Mondphasen 86
   13.4 Unordnung 91
14. Artikelverwaltung 93
15. ADREKART 113

Vorwort und Einführung 1

Der APPLE II und seine Nachfolger APPLE IIe und APPLE IIc sind überaus populäre und weit verbreitete Computer.

Dieser Rechner ist seit 1978 auf dem Markt und seit dieser Zeit ist sehr viel Software für diesen Rechner entwickelt worden. Im Gegensatz zu anderen Home Computer Systemen ist beim APPLE Wert darauf gelegt worden, daß die Programme kompatibel blieben. Ein Beispiel ist das Programm ADREKART aus diesem Buch. Es wurde 1979 für einen APPLE II mit DOS 3.2 geschrieben und läuft ohne Änderung nach Umsetzen auf DOS 3.3 auf dem APPLE IIc.

Die in diesem Buch zusammengestellten Programme bilden das Gerüst für die Entwicklung eigener Anwendungen. Zu Anfang wird gezeigt, daß es durchaus Alternativen zu der herkömmlichen Menütechnik gibt. Die Steuerung eines Programms durch Worte ist für viele ungeübte Computerbenutzer einfacher, als die Buchstabenauswahl aus einem Menü.

Die Verschiedenartigkeit von Programmen macht es notwendig, auch mit verschiedenen Datenstrukturen zu arbeiten. Deshalb werden in diesem Buch die Anwendung von verketteten Listen und von binären Bäumen gezeigt. Zwei Programme, NOTIZBUCH und KARTEI, haben im Prinzip den gleichen Programmablauf. Das einen Programm verwendet doppelt verkettete Listen, das andere binäre Bäume. Das Programm AVL zeigt, wie optimal ausbalancierte binäre Bäume erzeugt werden können.

Der APPLE II hatte von Anfang an eine ausgezeichnete hoch auflösende Grafik. Zwei Programme zeigen, wie ebene und drei dimensionale Funktionen auf dem Bildschirm dargestellt werden. Andere Grafikprogramme sind zur reinen Unterhaltung gedacht.

Die letzten beiden, sehr langen Programme stellen Programme zur Datenverwaltung dar. Sie verwenden zur Datenspeicherung index-sequentielle Dateien. Beide Programme sind selbständige, lauffähige Programme, die aber leicht auf andere, eigene Bedürfnisse umgestellt werden können.

Dazu eine Anmerkung: Das reine Abtippen eines Programmes ist in den meisten Fällen eine unbefriedigende Lösung. Es ist besser, zu versuchen, die Programmidee zu verstehen und, unter Verwendung von Programmteilen, eigene Lösungen zu entwickeln.

Der Verfasser möchte folgenden Autoren für das Beisteuern von Programmen danken:

Herrn W.BAUER für die Programme KARTEI, POLYGONGRAFIK und IRRGARTEN, sowie Herrn WOLFGANG F. HAHN für das Programm MONDPHASEN.

Den Lesern wünsche ich viel Spaß beim Programmieren und Experimentieren.

Holzkirchen, Herbst 1984 Ekkehard Flögel

Aufteilen von Zeichenketten 2

Das Programm Registrierkasse in Abschnitt 5 benutzt das Eingabeformat

5 Schraubendreher 1.95

Das Programm STRDIV in Abbildung 1 zeigt, wie man die eingegebene Zeichenkette in einzelne Teile zerlegt.

I$ ist die eingegebene Zeichenkette. An diese werden zwei Leerzeichen angefügt. In Zeile 1020 wird das erste Leerzeichen gesucht. Nachdem es gefunden wurde, wird der linke Teil von I$ in W$(0) gespeichert und aus I$ gelöscht. Die Länge L wird auf die Länge der übriggebliebenen Zeichenkette geändert. Jetzt wird auf die gleiche Weise das nächste Leerzeichen gesucht. Der neue linke Teil wird in W$(1) gespeichert. Danach wird der letzte Teil der Zeichenkette in W$(2) gespeichert.

```
1000  ONERR  GOTO 9000
1002  HOME : VTAB 2
1005  INPUT " ?";I$:V$ = I$:
      I$ = I$ + "  "
1010  K = 0:L =  LEN (I$):I = 1
1020  IF  MID$ (I$,I,1) <  > " " THEN I
      = I + 1: GOTO 1020
1030  W$(K) =  LEFT$ (I$,I - 1):
      I$ =  RIGHT$ (I$,L - I):
      K = K + 1:L = L - I:I = 1
1040  IF K = 3 THEN  GOTO 1100
1050  GOTO 1020
1100  N =  VAL (W$(0)):P =  VAL (W$(2))
1110  PRINT N: PRINT W$(1): PRINT P
```

```
1120 END
9000  VTAB 23: PRINT V$;" ?":
      GET A$
9010  GOTO 1002
```

Abbildung 1: Programm STRDIV.

Das Unterprogramm ab Zeile 9000 wird dann ausgeführt, wenn das Programm einen Fehler entdeckt. Die Eingabe von

```
5SCHRAUBENDREHER1.95
```

veranlasst das Unterprogramm

```
5SCHRAUBENDREHER1.95?
```

anzuzeigen. Nach Drücken einer beliebigen Taste springt das Programm zu Zeile 1000 zurück und wartet auf eine erneute Eingabe.

In Zeile 1100 werden die Zeichenketten W$(0) und W$(1) mit der VAL Funktion zu Zahlen umgerechnet und auf den Bildschirm ausgegeben.

Eine ähnliche Technik kann für Menüs benutzt werden. Das Programm MENU in Abbildung 2 zeigt ein Beispiel. Unser Menü hat drei Worte :

```
EINGABE
SCHREIBEN
ENDE
```

Diese Worte werden im Zeichenkettenfeld K$ gespeichert. Das Menü wird in Zeile 500-540 angezeigt. Die Eingabe wird in Zeile 1020 erwartet. Zeile 1060 sucht nach Leerzeichen. Wird kein Leerzeichen gefunden, so ist die Eingabe ein einzelnes Wort. Dieses Wort wird im im Zeichenkettenfeld K$ gesucht. Wird es gefunden, so springt das Programm zu der im ON..GOTO Befehl angegebenen Zeilennummer. Wird es nicht gefunden, dann wird die Eingabe über die Fehler Routine in Zeile 9000 angezeigt.

Falls in Zeile 1060 ein Leerzeichen gefunden wird, so wird die Eingabe in die zwei Zeichenketten V$ und U$ aufgeteilt. Jetzt wird V$ im Verzeichnis K$ gesucht. Wird es gefunden, dann springt das Programm zur vorgegebenen Zeilennummer. Hier wird der zweite Teil U$ benutzt, um im Programm zu verzweigen.

```
 100  K$(1) = "EINGABE"
 110  K$(2) = "SCHREIBEN"
 120  K$(3) = "ENDE"
 130  GOTO 1000
 500  HOME : VTAB 5: HTAB 10:
      PRINT "EINGABE"
 510  HTAB 10: PRINT "SCHREIBEN DISKETTE"
 520  HTAB 10: PRINT "SCHREIBEN LISTE"
 530  HTAB 10: PRINT "ENDE"
 540  RETURN
1000  GOSUB 500
1020  VTAB 17: HTAB 1: PRINT "?";:
      INPUT " ";I$
1030  VTAB 17: HTAB 1: PRINT L$;
1040  L =  LEN (I$)
1050  FOR I = 1 TO L
1060    IF  MID$ (I$,I,1) = " " THEN  GOTO 1100
1070  NEXT I
1080  V$ = I$: GOTO 1110
1100  V$ =  LEFT$ (I$,I - 1):
      U$ =  RIGHT$ (I$,L - I)
1110  FOR I = 1 TO 3
1120    IF V$ = K$(I) THEN  GOTO 1200
1130  NEXT I
1140  GOTO 9000
1200  ON I GOTO 2000,3000,4000
2000  REM   EINGABE
2010  VTAB 17: HTAB 1: PRINT "START EINER
      EINGABE-ROUTINE"
2020  GET A$: GOTO 1000
3000  REM  WRITE
3005  VTAB 17: HTAB 1
3010  IF U$ = "DISKETTE" THEN  PRINT "ICH
      SCHREIBE AUF DISKETTE": GOTO 2020
```

```
3020  IF U$ = "LISTE" THEN  PRINT "ICH DRUCKE
      EINE LISTE": GOTO 2020
3030  GOTO 9000
4000  REM  END
4010 END
9000  H = 0:V = 17: GOSUB 500:
      PRINT V$;" ?";
9005  GET I$:H = 0:V = 17: GOSUB 500:
      PRINT L$;: GOTO 1020
```

Abbildung 2: Programm MENU

Bei dieser Menü Technik müssen mehr Zeichen eingegeben werden, als bei der herkömmlichen Methode. Es zeigt sich aber, daß Computerbenutzer, die an das Schreiben mit einer Schreibmaschine gewöhnt sind, lieber einige Zeichen mehr eingeben, als einen Buchstaben aus dem Menü herauszusuchen.

Verkettete Listen

3

Betrachtet man Adressverwaltungen, Lagerverwaltungen oder andere Datenbankprogramme genauer, so wird man feststellen, daß alle Datensätze getrennte Einheiten sind. Sie haben keine Beziehung zum Vorgänger oder zum Nachfolger. Solche Dateien zu sortieren, heißt viele Bytes im Speicher zu bewegen, auch wenn man index-sequentielle Dateien benutzt.

Sehen wir uns ein Beispiel an:

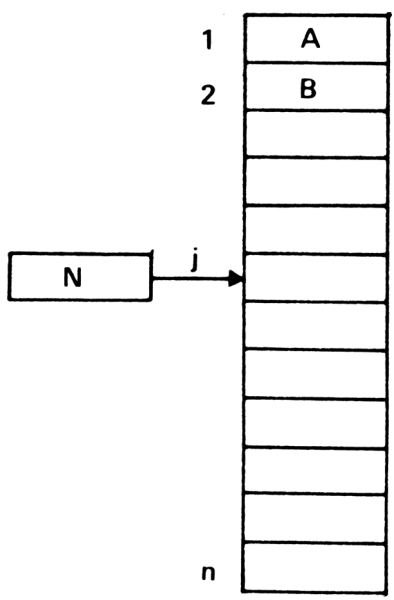


Abbildung 3: Einfügen einer Index-Nummer

Eine index-sequentielle Datei besteht aus 400 Datensätzen. Ein neuer Datensatz soll als 150. eingegeben werden. Für jeden Index benötigt man vier Bytes. Um für eine Index-nummer Platz zu machen, müssen 4 * 250 = 1000 Bytes bewegt werden.

Dies zeigt Abbildung 3.

Die Ausführungszeit von Sortierprogrammen wird von der Anzahl der zu bewegenden Bytes bestimmt. Dieses Bewegen kann man durch Benutzung verketteter Listen umgehen.

3.1 LINEARE LISTEN

Neben der Information, die im Datensatz gespeichert ist, enthält die verkettete Liste einen Zeiger. Dieser Zeiger ist die Adresse des nächsten Datensatzes. Zeiger sind Datentypen, die in den Sprachen PASCAL oder C benutzt werden. Abbildung 4 zeigt eine lineare verkettete Liste.

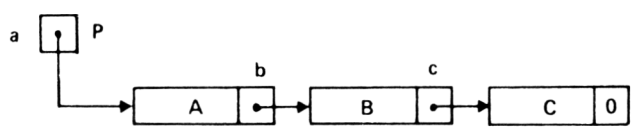


Abbildung 4: Verbundene lineare Liste.

Die Variable p in Abbildung 4 ist eine Zeiger-Variable. Sie enthält die Adresse des ersten Datensatzes. Dies ist die Adresse a des Datensatzes A. Der Zeiger von A ist die Adresse b des Datensatzes B. Der Zeiger des letzten Datensatzes ist Null. Dies markiert das Ende der Liste.

Das Einfügen eines neuen Datensatzes wird in Abbildung 5 gezeigt. Der Datensatz J soll zwischen Datensatz A und Datensatz B eingefügt werden.

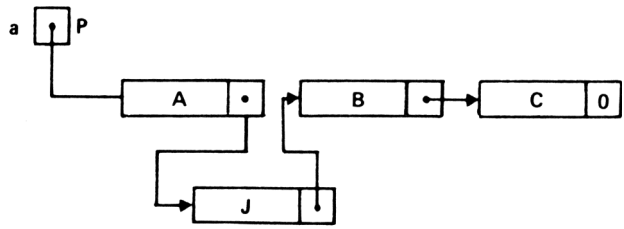


ABBILDUNG 5 : Einfügen eines neuen Datensatzes.

Der Datensatz J wird irgendwo im Speicher unter Adresse j abgelegt. Diese Adresse wird in den Zeiger von A geschrieben. Der vorherige Zeiger von A (b) wird in den Zeiger von J geschrieben. Der Datensatz J passt jetzt logisch in die verkettete Liste obwohl er an anderer Stelle abgespeichert ist.

Das Programm LILI in Abbildung 6 ist ein Demonstrationsprogramm für lineare verkettete Listen. Zeile 100 dimensioniert die beiden Felder N$ und Z.

Die Länge der beiden Felder wird auf 10 festgesetzt. Im Feld N$ werden die Daten (Namen, Sicherungsnummern, Artikel usw) abgespeichert. Das Feld Z enthält den Zeiger. Das Programm erwartet Dateneingabe in den Zeilen 500-540. Ist die Eingabe beendet, verbindet das Programm die Einträge. Dies wird in den Zeilen 200 bis 299 ausgeführt.

Die Zeigervariable P ist am Anfang Null, als Anzeige für eine leere Liste. Das Verketten beginnt mit P=1 und Z(1)=0. In Zeile 250 wird bestimmt, ob der neue Datensatz vor den ersten Eintrag gestellt wird. Zeile 270 stellt den neuen Datensatz zwischen zwei existierende, während ihn Zeile 280 an das Ende der Liste stellt.

```
100  N = 10: DIM N$(N),Z(N)
110  GOSUB 500
120  GOSUB 200
140  GOSUB 1000
150  PRINT : PRINT
```

```
160  GOSUB 1020
170  STOP
200  REM  LINK LIST
210  P = 0
220  IF P = 0 THEN Z(1) = 0:
     P = 1
230  FOR J = 2 TO N
240    I = P
250    IF N$(J) < N$(P) THEN Z(J) = P:
       P = J: GOTO 295
260    K = I:I = Z(I)
270    IF N$(J) < N$(I) THEN Z(K) = J:
       Z(J) = I: GOTO 295
280    IF I = 0 THEN Z(K) = J:
       Z(J) = 0: GOTO 295
290    GOTO 260
295  NEXT J
299  RETURN
500  REM  INPUT
510  FOR J = 1 TO N
520    INPUT ":";N$(J)
530  NEXT J: RETURN
1000 FOR I = 1 TO N: PRINT N$(I):
     NEXT
1010 RETURN
1020 I = P
1030 IF I = 0 THEN  RETURN
1040 PRINT N$(I)
1050 I = Z(I): GOTO 1030
```

Abbildung 6: Lineare verkettete Listen.

Die Zeilen 1000-1010 drucken die Liste in der eingegebenen Ordnung. Mit den Zeilen 1020-1050 wird die sortierte verkettete Liste gedruckt. Abbildung 7 zeigt ein Beispiel.

MUELLER GEORG	ADLER HORST
WEDIG HANS	BAUM EDGAR
STERN PETER	BECKER KARL
ADLER HORST	DUCK DONALD
JUERGENS UDO	JUERGENS UDO
LINDENBERG UDO	LINDENBERG UDO
ZELLER MANFRED	MUELLER GEORG
DUCK DONALD	STERN PETER
BECKER KARL	WEDIG HANS
BAUM EDGAR	ZELLER MANFRED

Abbildung 7: Beispiel verkettete Liste.

In Abbildung 7 stehen die Namen auf der linken Seite wie sie eingegeben wurden. Der Zeiger P zeigt auf den vierten Namen ADLER HORST. Der Zeiger davon zeigt auf Name Nummer 10. Auf der rechten Seite steht eine sortierte Liste. Der letzte Name, ZELLER MANFRED, hat den Zeiger 0. Dies ist das Ende der Liste.

Einen Datensatz zu löschen ist auch ganz einfach, wie Abbildung 8 zeigt. Der Datensatz B soll aus der Liste gelöscht werden. Der Zeiger von B(c) wird in den Zeiger von A geschrieben.

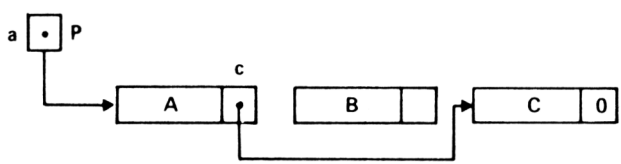


Abbildung 8: Löschen eines Datensatzes.

3.2 Doppelt verkettete Listen.

Abbildung 9 zeigt eine doppelt verbundene Liste. Jeder Datensatz hat zwei Zeiger. Einer zeigt auf den Vor-

gänger, der andere auf den Nachfolger. Eine doppelt verkettete Liste kann in beiden Richtungen verarbeitet werden. Der linke Zeiger des ersten Datensatzes und der rechte Zeiger des letzten Datensatzes sind Null. Das Programm NOTIZBUCH benutzt solch eine doppelt verkettete Liste.

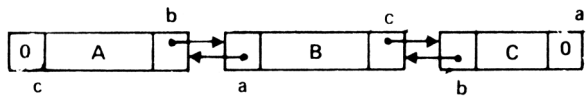


Abbildung 9: Doppelt verkettete Liste.

Hashing 4

Ein Hauptproblem in der Datenbank-Verwaltung besteht darin, in sehr kurzer Zeit einen Datensatz in einer großen Datei zu finden. Diese Forderung nach einer kurzen Zugriffszeit in einer großen Datenbank scheint unrealistisch zu sein. Anstatt zu Sortieren und zu Suchen kann man HASHING benutzen.

Hier ein Beispiel für dieses Verfahren. Abbildung 10 zeigt eine sequentielle Liste von Namen. Die Länge der Namen kann variieren, doch die Länge des Datensatzes ist konstant. Die Datensätze sind nicht sortiert.

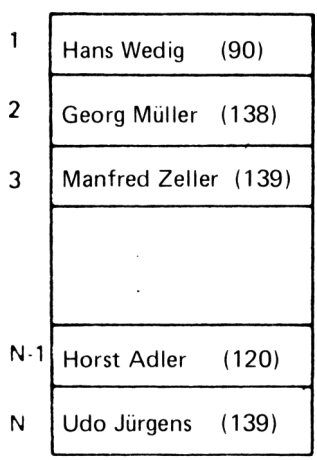


Abbildung 10: Namensliste.

Neben dieser Liste wird eine zweite Liste benutzt. Diese Liste enthält Zeiger auf die Datensätze. Diese Zeiger können auf verschiedene Art gefunden werden.

Eine Möglichkeit ist zum Beispiel, den Zeichen Zahlen zuzuordnen. "A" entspricht 1 und "Z" entspricht 26. Die kleinste Nummer in diesem Beispiel ist 1. Der Name besteht nur aus dem Buchstaben A. Die größte Zahl ist 520 (20 * Z). Die Summe der Buchstaben des ersten Namens HANS WEDIG ist 90. Die Summe des zweiten Namens GEORG MUELLER ist 138. Dies kann mit dem Programm in Abbildung 11 ausgerechnet werden.

```
 10  INPUT "NAME: ";A$
 20  GOSUB 100
 30  PRINT S
 40  INPUT "NOCHMAL ( /N)";I$
 50  IF I$ = "N" THEN END
 60  GOTO 10
100  L =  LEN (A$)
110  S =  ASC ( LEFT$ (A$,1)) - 64
120  FOR I = 2 TO L - 1
130    N =  ASC ( MID$ (A$,I,1))
140    IF N < 64 GOTO 170
150    IF N > 90 GOTO 170
160    S = S + N - 64
170  NEXT I
180  S = S +  ASC ( RIGHT$ (A$,1)) - 64
190  RETURN
```

```
 RUN
NAME: HANS WEDIG
90
NOCHMAL ( /N)
NAME: GEORG MUELLER
138
NOCHMAL ( /N)
NAME: MANFRED ZELLER
139
NOCHMAL ( /N)
NAME: HORST ADLER
120
NOCHMAL ( /N)
NAME: UDO JUERGENS
139
NOCHMAL ( /N)N
```

Abbildung 11:
Errechnen der HASH-Nummern mit der Buchstabensumme.

Abbildung 12 zeigt die Verbindung zwischen der Namensliste und der zweiten Liste. Am Anfang sind alle Felder der Zeiger-Liste Null. Die HASH-Nummer des ersten Namens war 90. Deshalb wird in das 90. Feld der Zeigerliste eine 1 geschrieben. Das 138. Feld erhält die Zahl 2

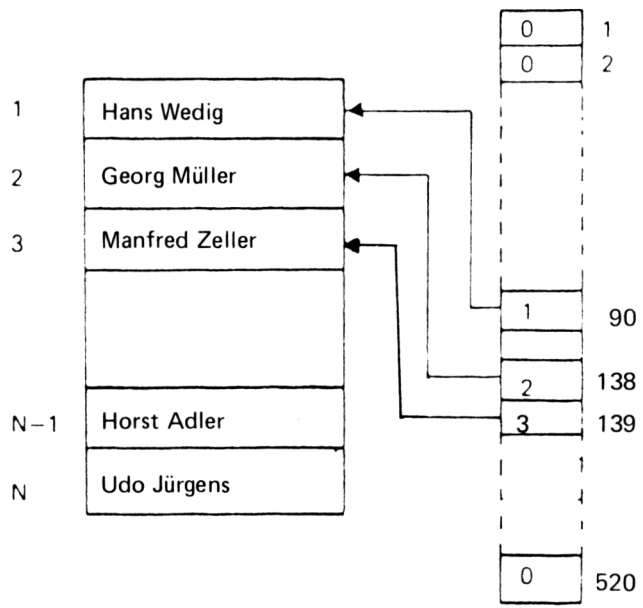


Abbildung 12: Namensliste und Zeiger-Liste.

Jetzt ist es sehr leicht einen Namen zu finden. Will man GEORG MUELLER finden, muß man nur den HASH-Wert berechnen. Diese Zahl gibt die Nummer des Feldes an, wo der Zeiger auf diesen Namen gespeichert ist. Dies ist sehr einfach und erlaubt einen schnellen Zugriff auf den Datensatz. Sehen wir uns nun zwei andere Namen an. MANFRED ZELLER und UDO JUERGENS. Beide Namen haben, mit der obigen Rechen-Methode, die HASH-Nummer 139. Es liegt eine Kollision vor. Nun besteht Problem darin, eine Methode zu finden, die weniger Kollisionsmöglichkeiten bietet.

Andere Methoden:

- Nummer des ersten Buchstabens plus Nummer des letzten Buchstabens mal Länge des Namens.
- Nummer des ersten Buchstabens plus Nummer des zweiten Buchstabens hoch zwei plus Nummer des dritten Buchstabens hoch drei.
- Benutzung einer anderen Nummernverteilung.
- nur Konsonanten addieren.

Aber es kann in allen Fällen zu Kollisionen kommen. Es gibt mehrere Methoden um mit diesen Kollisionen zu arbeiten. Eine dieser Methoden wird hier gezeigt. Zu der Namensliste und der Zeigerliste wird eine dritte, zweidimensionale Liste hinzugefügt. Dies zeigt Abbildung 13.

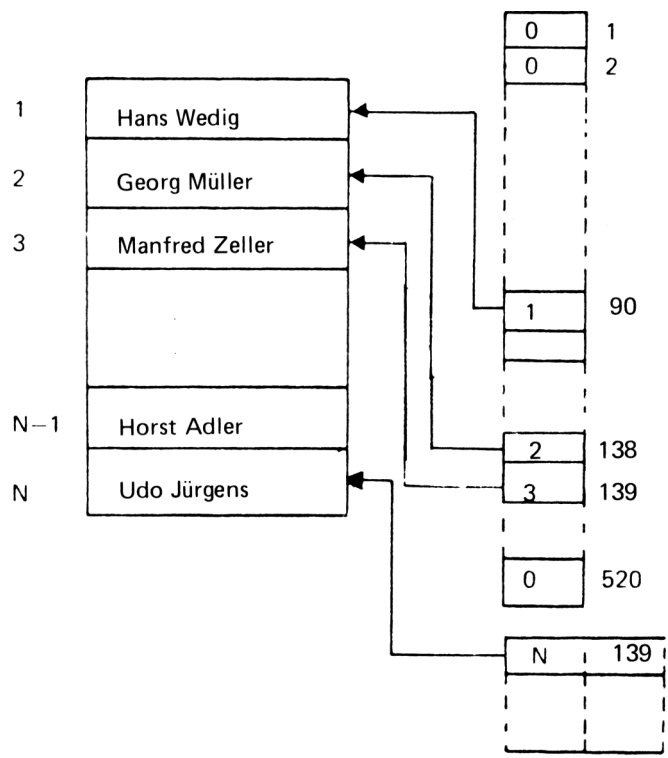


Abbildung 13: Arbeiten mit Kollisionen.

Wird eine Kollision entdeckt, so werden die Nummer des Eintrags und die HASH-Nummer in der dritten Liste gespeichert. Um einen Namen zu finden muss man zwei Schritte durchführen. Der erste Schritt ist, den Namen über die Zeiger-Liste zu finden. Ist es nicht der gewünschte Namen, so wird der richtige Name in der dritten Liste gesucht.

Bei einigen Methoden ist die HASH-Nummer sehr groß. Dann sollte man die MODULO-Division benutzen um eine Zahl mit einer Obergrenze zu bekommen. Bei einer Liste von 300 Namen wird eine HASH-Nummer zwischen 0 und 300 erwartet. Dies wird mit einer MODULO-Division durchgeführt.

```
Q = INT(S/300)
R = (S/300-Q)*300
```

Q ist der Quotient und R der Rest der Division. R ist immer größer oder gleich Null und kleiner als 300.

Ein vollständiges BASIC-Programm wird hier nicht angegeben. Schreiben Sie ein eigenes Programm und testen Sie es mit verschiedenen Datentypen.

Notizen

Registrierkasse 5

```
                LADENKASSE
DATUM:27.08.84              SUMME:     6.85
-----------------------------------------


                                        1.85/1

                                        2.10/3

                                        1.05/5
 ? DATUM 27.08.84
 ? VERKAUF
 ? 2 H-MILCH 1.05 3                     1.85/8
 ? 1 TEEWURST 1.85 1
 ? 3 SEMMELN 0.35 5
 ? 1 COCA-COLA 1.85 8
 ?
-----------------------------------------
MENGE BEZEICHNUNG PREIS KONTO
Z.B. 5 WEINGLAESER 3.55 3
MIT CTRL-A ENDE DER EINGABE
```

Abbildung 14: Bildschirmausgabe Registrierkasse.

QUICKY-MARKT 27.08.84
INH. FRANZ MUELLER
GRUENSTR.16
8000 MUENCHEN 1

QUITTUNG

2	H-MILCH	1.05	2.10
1	TEEWURST	1.85	1.85
3	SEMMELN	0.35	1.05
1	COCA-COLA	1.85	1.85

BETRAG	6.85
MWST 14%	.96
BEZAHLT	10.00
RUECKGABE	3.15

Abbildung 15: Ausdruck einer Quittung.

```
 105  D$ =  CHR$ (13) +  CHR$ (4)
 110  DIM K$(7),S$(15),GP(10)
 115  DIM N$(4): GOSUB 7500
 120  K$(1) = "DATUM":K$(2) = "LESE":
      K$(3) = "SPEICHERN":
      K$(4) = "ENDE":K$(5) = "VERKAUF"
 125  K$(6) = "QUITTUNG"
 130  L$ = "                    ":
      REM   20 SPACES
 140  BO = 0
 199  GOTO 1000
 520  N$(1) = "INH. FRANZ MUELLER"
 902  POKE 37,V: POKE 36,H:
      CALL  - 990: RETURN
1000  HOME : HTAB 10: PRINT "LADENKASSE"
1010  VTAB 2: PRINT "DATUM:
      ";: HTAB 25: PRINT "SUMME:
      "
1012  VTAB 3:
      FOR I = 0 TO 39: PRINT "-";:
      NEXT
```

```
1014  VTAB 19:
      FOR I = 0 TO 39: PRINT "-";:
      NEXT
1015  GOSUB 1500
1020  V = 17:H = 1: GOSUB 900:
      PRINT "?";: INPUT " ";I$
1030  V = 17:H = 1: GOSUB 900:
      PRINT L$;
1040  L =  LEN (I$)
1050  FOR I = 1 TO L
1060    IF  MID$ (I$,I,1) = " " THEN  GOTO 1100
1070  NEXT I
1080  V$ = I$: GOTO 1110
1100  V$ =  LEFT$ (I$,I - 1):
      U$ =  RIGHT$ (I$,L - I)
1110  FOR I = 1 TO 6
1120    IF V$ = K$(I) THEN  GOTO 1200
1130  NEXT I
1140  GOTO 9000
1200  ON I GOTO 2000,3000,4000,5000,6000,
      7000
1500  POKE 34,19: CALL  - 958:
      H = 0:V = 19: GOSUB 900
1502  CALL  - 958
1504  PRINT "DATUM TT.MM.JJ, VERKAUF, ENDE"
1505  PRINT "LESE <DATUM>, SPEICHERN "
1507  PRINT "QUITTUNG"
1510  POKE 33,30< POKE 34,3:
      POKE 35,18
1520  RETURN
1600  P$ =  STR$ (P):L =  LEN (P$)
1610  FOR I = 1 TO L
1620    IF  MID$ (P$,I,1) = "." THEN  GOTO 1640
1630  NEXT I:P$ = P$ + ".00"
1640  IF I = L - 1 THEN P$ = P$ + "0"
1650  RETURN
2000  REM  INPUT DATE
2010  DA$ = U$: VTAB 2: HTAB 7:
      PRINT U$: GOTO 1020
3000  REM  READ DISK
3005  VTAB 2: HTAB 7: PRINT U$:
      PRINT :DA$ = U$
```

```
3010  U$ = "D" + U$
3020  PRINT D$;"OPEN";U$
3030  PRINT D$;"READ";U$
3040  INPUT T
3050  FOR I = 0 TO 9: INPUT GP(I):
      NEXT
3060  PRINT D$;"CLOSE";U$
3065  POKE 33,40
3070  H = 0:V = 4: GOSUB 900:
      HOME
3072  P = T: GOSUB 1600
3075  V = 1:L =  LEN (P$):
      H = 38 - L: GOSUB 900
3077  PRINT P$
3080  FOR J = 0 TO 9
3090    P = GP(J): GOSUB 1600
3100    V = J + 5:L =  LEN (P$):
        H = 38 - L: GOSUB 900
3110    PRINT P$;"/";J
3120  NEXT J
3130  POKE 33,30: GOTO 1020
4000  REM  WRITE
4100  U$ = "D" + DA$: PRINT
5134  PRINT D$;"OPEN";U$
4120  PRINT D$;"DELETE";U$
4130  PRINT D$;"OPEN";U$
4140  PRINT D$;"WRITE";U$
4150  PRINT T
4160  FOR I = 0 TO 9: PRINT GP(I):
      NEXT
4170  PRINT D$;"CLOSE";U$
4175  IF BO = 1 THEN 5005
4180  GOTO 1020
5000  BO = 1: GOTO 4000
5005  POKE 33,40: POKE 34,0:
      POKE 35,24
5010  HOME :END
6000  REM  INPUT
6010  POKE 35,24: POKE 34,19:
      H = 0:V = 19: GOSUB 900
6012  CALL  - 958
6020  PRINT "MENGE BEZEICHNUNG PREIS KONTO"
6030  PRINT "Z.B. 5 WEINGLAESER 3.55 3"
```

```
6040  PRINT "MIT CTRL-A ENDE DER EINGABE"
6045  J = 0
6050  I$ = "": POKE 34,3:
      POKE 35,18
6055  V = 17:H = 1: GOSUB 900
6060  PRINT "?";: INPUT " ";I$:
      I$ = I$ + "  "
6065  S$(J) = I$:J = J + 1:
      IF J > 14 THEN 9000
6070  IF  ASC (I$) = 1 THEN  POKE 35,24:
      GOTO 1014
6080  GOSUB 6300
6120  N =  VAL (W$(0)):P =  VAL *W$(2)):
      G =  VAL (W$(3))
6130  P = P * N:P0 = P:P = P + GP(G):
      GP(G) = P: GOSUB 1600
6135  POKE 33,40
6140  V = G + 5:L =  LEN (P$):
      H = 38 - L: GOSUB 900:
      PRINT P$;"/";G
6150  P = T + P0:T = P: GOSUB 1600
6160  V = 1:L =  LEN (P$):
      H = 38 - L: GOSUB 900:
      PRINT P$;
6170  POKE 33,30: GOTO 6050
6200  H = 0:V = 17: GOSUB 900
6210  J = J - 1: PRINT S$(J);" ?":
      GET A$
6220  GOTO 6055
6300  K = 0:L =  LEN (I$):
      I = 1
6310  IF  MID$ (I$,I,1) <  > " " THEN I
      = I + 1: GOTO 6310
6320  W$(K) =  LEFT$ (I$,I - 1):
      I$ =  RIGHT$ (I$,L - I):
      K = K + 1:L = L - I:
      I = 1
6330  IF K = 4 THEN  RETURN
6340  GOTO 6310
7000  REM  WRITE RECIEPT
7010  H = 0:V = 4: GOSUB 900:
      HOME
7020  H = 1:V = 17: GOSUB 900
```

```
7030  INPUT "MWST: ";T$
7035  PRINT D$;"PR#2"
7040  VP = 0
7042  VT = 0:A$ = N$(0):
      GOSUB 7400
7044  VT = 54:A$ = DA$: GOSUB 7400
7046  PRINT :VP = 0
7050  VT = 0:A$ = N$(1):
      GOSUB 7400
7055  PRINT :VP = 0
7060  VT = 0:A$ = N$(2):
      GOSUB 7400
7065  PRINT :VP = 0
7070  VT = 0:A$ = N$(3):
      GOSUB 7400
7075  PRINT :VP = 0
7080  VT = 30:A$ = "QUITTUNG":
      GOSUB 7400
7085  PRINT :VP = 0
7090  T = 0
7100  FOR Z = 0 TO J - 2
7110    I$ = S$(Z): GOSUB 6300
7120    PRINT :VP = 0
7130    VT = 0:VF = 5:A$ = W$(0):
        GOSUB 7600
7140    VT = 10:A$ = W$(1):
        GOSUB 7400
7150    VT = 30:VF = 10:
        A$ = W$(2): GOSUB 7600
7160    N =  VAL (W$(0)):
        P =  VAL (W$(2)):P = N * P:
        T = T + P
7170    GOSUB 1600
7180    VT = 40:VF = 10:
        A$ = P$: GOSUB 7600
7190  NEXT Z
7200  PRINT : PRINT :VP = 0
7210  VT = 25:A$ = "BETRAG":
      GOSUB 7400
7220  P = T: GOSUB 1600:
      VT = 40:VF = 10:A$ = P$:
      GOSUB 7600
7230  PRINT :VP = 0
```

```
7240  VT = 25:A$ = "MWST":
      GOSUB 7400:VT = 31:
      A$ = T$ + "%": GOSUB 7400
7245  TA =  VAL (T$):P =  INT (T * TA + .5)
      / 100
7250  GOSUB 7390
7255  P = T
7260  PRINT D$;"PR#0"
7270  H = 1:V = 17: GOSUB 900
7280  INPUT "BEZAHLT";A$
7290  P1 =  VAL (A$): IF P1 < P THEN  GOTO 7270
7300  H = 1:V = 17: GOSUB 900
7310  P2 = P1 - P:P = P2:
      GOSUB 1600
7320  PRINT "RUECKGABE";P$
7325  PRINT D$;"PR#2"
7328  PRINT :VP = 0
7330  VT = 25:A$ = "BEZAHLT":
      GOSUB 7400
7340  P = P1: GOSUB 7390
7345  PRINT :VP = 0
7350  VT = 25:A$ = "RUECKGABE":
      GOSUB 7400
7360  P = P2: GOSUB 7390
7365  PRINT D$;"PR#0"
7370  GOTO 1020
7390  GOSUB 1600:VT = 40:
      VF = 10:A$ = P$: GOTO 7600
7400  L =  LEN (A$)
7410  IF VT = VP THEN  PRINT A$;:
      VP = L: GOTO 7499
7430  IF VT - VP > 20 THEN  PRINT L$;:
      VP = VP + 20: GOTO 7410
7440  A$ =  LEFT$ (L$,VT - VP) + A$:
      PRINT A$;:VP = VT + L
7499  RETURN
7500  REM  SALES ADDRESS
7510  N$(0) = "QUICKY-MARKT"
7520  N$(1) = "INH. FRANZ MUELLER"
7530  N$(2) = "GRUENSTR.16"
7540  N$(3) = "8000 MUENCHEN 1"
7550  RETURN
```

```
7600  L =  LEN (A$): IF L > VF THEN VT = VT
      - (L - VF)
7610  IF VP > VT + VF THEN  PRINT A$;:
      VP = VP + L: GOTO 7699
7620  IF VT - VP > 20 THEN  PRINT L$;:
      VP = VP + 20: GOTO 7620
7630  IF VT - VP <  > 0 THEN  PRINT  LEFT$ (L$,
      VT - VP);:VP = VT
7640  A$ =  LEFT$ (L$,VF - L) + A$:
      PRINT A$;:VP = VP + VF
7699  RETURN
9000  H = 0:V = 17: GOSUB 900:
      PRINT V$;" ?";
9005  H = 0:V = 17: GOSUB 900:
      PRINT L$;: GOTO 1020
```

Abbildung 16: Programm Registrierkasse.

Programmmenü.

DATUM TT.MM.JJ	Eingabe des Datums
VERKAUF	Eingabe von verkauften Artikeln.
LESE <DATM>	Einlesen der Umsätze eines Tages.
SPEICHERN	Umsatz auf Diskette speichern.
QUITTUNG	Ausdrucken einer Quittung.

Oben rechts auf dem Bildschirm wird der Gesamtumsatz, rechts darunter der Umsatz in den einzelnen Konten angegeben. Als Kontonummern stehen die Zahlen 1 bis 9 zur Verfügung. Die Eingabe für einen Kunden wird mit CTRL-A abgeschlossen. Danach kann mit dem Wort QUITTUNG eine Quittung ausgeschrieben weren.

Das Wort "SPEICHERN" schreibt die bis dahin erzielten Umsätze auf Diskette. Sollen nach dem Abbruch des Programms neue Beträge zu einem Tagesumsatz hinzuaddiert werden, so werden nach "RUN" durch "LESE TT.MM.JJ" der Gesamtumsatz und die Umsätze in den einzelnen Konten eingelesen. Neue Eingaben werden dann dazu addiert.

Programm-Erläuterung.

Die Haupteingaberoutine beginnt in Zeile 1000. Ab Zeile 2000 wird das Datum eingegeben. Die Zeilen 3000-3130 lesen den Umsatz eines Tages von der Diskette und zeigen ihn auf dem Bildschirm an. In den Zeilen 4000-4180 wird der Umsatz eines Tages auf Diskette gespeichert.

Ist als Datum z.B. 15.05.84 eingegeben worden, dann wird die Datei mit dem Namen D15.05.84 versehen und so aufgezeichnet. Das "D" wird automatisch vorangestellt.

Die Zeilen 5000 bis 5010 beenden das Programm. Die Eingaben für den Verkauf werden in den Zeilen 6000 bis 6340 aufgeführt. Bis zu 15 einzelne Posten können eingegeben werden. Diese Eingaben werden in dem Zeichenkettenfeld S$ gespeichert.

Das Unterprogramm ab Zeile 6300 zerlegt die Eingabezeile in die vier einzelnen Teile. Wenn bei der Eingabe ein Fehler gemacht wird, springt das Programm in eine Fehlerroutine ab Zeile 6200.

Das Unterprogramm ab Zeile 1600 wird zur formatierten Ausgabe von Zahlen verwendet. Der Preis wird als Variable P an diese Programm übergeben. Diese Zahl wird in eine Zeichenkette P$ gewandelt. Dann wird nach dem Dezimalpunkt gesucht und, falls keiner vorhanden ist, wird 00 an diese Zeichenkette angehängt. Ist ein Dezimalpunkt vorhanden, so wird geprüft, ob zwei Stellen darauf folgen. Wenn nicht, wird eine Null angehängt.

Der Ausdruck einer Quittung beginnt in Zeile 1000. Die Geschäftsadresse ist im Zeichenkettenfeld N$(0) bis N$(4) gespeichert. Diese Adresse, das Datum und das Wort Quittung werden ausgedruckt. Dann wird der Inhalt des Zeichenkettenfeldes S$ ausgedruckt und die errechneten Beträge werden addiert. Am Ende der Quittung wird der Gesamtbetrag und der darin enthaltene Anteil der Mehrwertsteuer ausgedruckt. Der gezahlte Betrag wird eingeben, das Rausgeld ausgerechnet und ebenfalls ausgedruckt.

Für das Formatieren der Druckausgabe werden zwei Unterprogramme verwendet. Das Unterprogramm ab Zeile 1400 druckt eine Zeichenkette linksbündig ab der horizontalen Position VT aus. Am Beginn einer neuen Zeile ist der Wert der Variablen VP Null. Danach enthält sie die augenblickliche Druckposition. Die Länge einer Bezeichnung sollte nicht länger als 20 Zeichen sein.

Das Unterprogramm ab Zeile 1600 druckt eine Zeichenkette rechtsbündig in einem Feld der Breite VF aus. Der Beginn dieses Feldes ist in der Variablen VT enthalten. Die darin enthaltenen POKE-Befehle und Aufrufe von Maschinenprogrammen betreffen die Aufteilung des Bildschirms.

Notizbuch 6

Das Programm NOTIZBUCH in Abbildung 18 benutzt doppelt verkettete Listen. Für die Zeiger R(i) und L(i) werden zwei Felder benutzt. Der Zeiger R(i) zeigt auf das nächste größere Schlüsselwort und L(i) zeigt auf das kleinere Schlüsselwort. Ein Eintrag steht logisch rechts vom aktuellen Eintrag, der andere steht links vom Eintrag. Im Speicher stehen sie an verschiedenen Plätzen.

Bei der Aufstellung einer doppelt verketteten Liste gibt es fünf Möglichkeiten:

1. Die Liste ist leer. Dann ist P=0.

2. Es gibt nur einen Eintrag. Dann ist P=1, R(1)=0 und L(1)=0.

3. Der nächste Eintrag j ist größer als alle Einträge in derListe. Dann sind R(j)=0, L(j)=k, R(k)=j. K ist die Nummer des früheren letzten Eintrags.

4. Der nächste Eintrag j ist kleiner als alle Einträge.Dann sind R(j)=p, L(j)=0, L(p)=j und P=j.

5. Der nächste Eintrag muss zwischen Eintrag k und i gestellt werden. Dann sind R(k)=j, R(j)=i, L(j)=k und L(i)=j.

Dieses Verbinden ist in den Zeilen 200-299 programmiert. Abbildung 17 zeigt den Aufbau des Notizbuches.

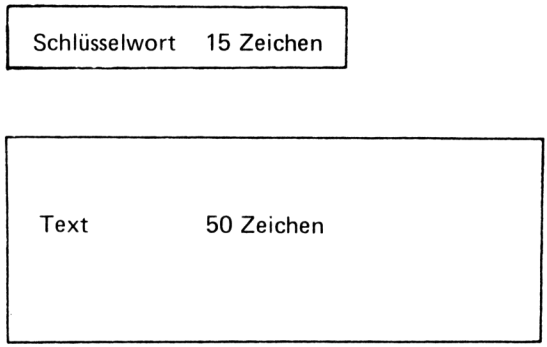


Abbildung 17: Aufbau des Notizbuches.

Ein Schlüsselwort kann aus bis zu 15 Zeichen bestehen und ein Eintrag kann bis zu 60 Zeichen lang sein. In den Eingabemodus kommt man durch das Wort EINTRAG. Als Beispiel werden wir Städte als Schlüsselworte benutzen und in dem Notizbuch einige Bemerkungen über diese Städte machen.

Zum Beispiel:

```
Schlüsselwort: STUTTGART
Text  :HANS MEIER TEL 123456 ANRUFEN
```

Im unteren Teil des Bildschirms erscheint das Zeichen -->?. Drückt man die rechte Cursor-Taste, dann kann man einen weiteren Eintrag machen. Jede andere Taste veranlasst das Programm ins Menue zurück zu springen.

Um einen Eintrag im Notizbuch zu finden muss man folgendes eingeben:

```
SUCHE <SCHLüSSELWORT>
```

Zum Beispiel:

```
SUCHE STUTTGART
```

Das Schlüsselwort und der Text werden auf dem Schirm angezeigt. Jetzt erscheinen die Zeichen <-->? im unteren

Bildschirmteil. Die linke Cursor-Taste wird benutzt um zum nächstkleineren Schlüsselwort zu gelangen und die rechte Cursor-Taste um zum nächstgrößeren Schlüsselwort im Notizbuch zu gelangen. Die Meldung ENDE DER LISTE wird gedruckt, falls eines der beiden Enden der Liste erreicht ist. Das Drücken irgendeiner anderen Taste außer den Cursor-tasten stoppt das Durchblättern des Notiz- und führt zurück zum Menü.

Das Wort DRUCKE gibt alle Schlüsselworte sortiert auf dem Bildschirm aus. Mit SCHREIBE <DATEINAME> kann das Notizbuch auf Diskette abgespeichert werden. Mit LESE <DATEINAME> wird das Notizbuch wieder in den Speicher geladen. Das Wort ENDE beendet das Programm.

```
100  N = 40:Z = 0:C = 0
105  DIM N$(N),R(N),L(N)
110  DIM K$(6),E$(N)
120  K$(1) = "EINTRAG":
     K$(2) = "SUCHE"
130  K$(3) = "ENDE"
140  K$(4) = "SCHREIBE":
     K$(5) = "LESE":K$(6) = "DRUCKE"
150  L$ = "                    ":
     REM  20 SPACES
160  D$ =  CHR$ (4)
199  GOTO 1000
200  REM  MAKE DOUBLE LINKED LISTS
210  R(1) = 0:L(1) = 0:
     P = 1
220  FOR J = 2 TO Z
230    I = P
250    IF N$(J) < N$(P) THEN R(J) = P:
       L(J) = 0:L(P) = J:P = J:
       GOTO 295
260    K = I:I = R(I)
265    IF I = 0 THEN R(K) = J:
       R(J) = 0:L(J) = K: GOTO 295
270    IF N$(J) < N$(I) THEN R(K) = J:
       R(J) = I:L(J) = K:L(I) = J:
       GOTO 295
290    GOTO 260
```

```
 295  NEXT J
 298  C = 1: RETURN
1000  HOME : VTAB 3: HTAB 5:
      PRINT "EINTRAG"
1005  HTAB 5: PRINT "SUCHE <SCHLUESSELWORT>"
1008  HTAB 5: PRINT "LESE <DATEINAME>"
1010  HTAB 5: PRINT "SCHREIBE <DATEINAME>"
1014  HTAB 5: PRINT "DRUCKE <SCHLUESSELWORTE>
1018  HTAB 5: PRINT "ENDE
1020  VTAB 20: INPUT " ? ";I$
1030  VTAB 20: PRINT L$
1040  L =  LEN (I$)
1050  FOR I = 1 TO L
1060    IF  MID$ (I$,I,1) = " " THEN  GOTO 1100
1070  NEXT I
1080  V$ = I$: GOTO 1110
1100  V$ =  LEFT$ (I$,I - 1):
      U$ =  RIGHT$ (I$,L - I)
1110  FOR I = 1 TO 6
1120    IF V$ = K$(I) THEN  GOTO 1200
1130  NEXT I
1140  GOTO 9000
1200  ON I GOTO 2000,3000,4000,5000,6000,
      7000
2000  REM  MAKE ENTRY
2005  HOME
2010  Z = Z + 1
2020  INPUT "SCHLUESSELWORT:
       ";N$(Z)
2030  INPUT "TEXT:";E$(Z)
2040  VTAB 20: PRINT "-->"
2045  C = 0
2050  GET I$: IF  ASC (I$) = 21 THEN  GOTO 2005
2060  GOTO 1000
3000  REM  SEARCH
3010  HOME : IF C = 0 THEN  PRINT "VERKETTUNG":
      GOSUB 200
3020  I = P
3030  IF I = 0 THEN  GOTO 3080
3040  IF N$(I) = U$ THEN  HOME :
      VTAB 2: PRINT N$(I):
      VTAB 4: PRINT E$(I):
      VTAB 20: PRINT "<-- -->":
      GOTO 3100
```

```
3050  I = R(I): GOTO 3030
3080  VTAB 20: PRINT I$;" ? ":
      GOTO 9005
3100  GET I$:
3110  IF  ASC (I$) = 21 THEN K = I:
      I = R(I): GOTO 3200
3120  IF  ASC (I$) = 08 THEN K = I:
      I = L(I): GOTO 3200
3130  GOTO 1000
3200  HOME : IF I = 0 THEN  GOTO 3240
3210  VTAB 2: PRINT N$(I):
      VTAB 4: PRINT E$(I):
      VTAB 20: PRINT "<— —>":
      GOTO 3100
3240  PRINT "ENDE DER LISTE"
3250  I = K: GET I$: GOTO 3210
4000  VTAB 20: PRINT "AUF DISKETTE SPEICHERN
       ( /N)"
4010  GET I$: IF I$ = "N" THEN END
4020  GOTO 1000
5000  REM  WRITE TO DISK
5003  HOME
5010  PRINT D$;"OPEN";U$
5020  PRINT D$;"DELETE";U$
5030  PRINT D$;"OPEN";U$
5040  PRINT D$;"WRITE";U$
5050  PRINT Z
5060  FOR I = 1 TO Z
5070    PRINT N$(I):
      NEXT I
5080  FOR I = 1 TO Z
5090    PRINT E$(I):
      NEXT I
5100  PRINT D$;"CLOSE";U$
5110  GOTO 1000
6000  REM  READ FROM DISK
6010  ONERR  GOTO 9000
6020  PRINT D$;"OPEN";U$
6030  PRINT D$;"READ";U$
6040  INPUT Z
6050  FOR I = 1 TO Z
6060    INPUT N$(I):
      NEXT I
```

```
6070  FOR I = 1 TO Z
6080    INPUT E$(I):
      NEXT I
6090  PRINT D$;"CLOSE";U$
6095  PRINT "VERKETTEUNG":
      GOSUB 200
6100  GOTO 1000
7000  REM  PRINT KEYWORDS
7005  IF C = 0 THEN  HOME :
      PRINT "VERKETTUNG ":
      GOSUB 200
7010  HOME :I = P:K = 0
7020  PRINT N$(I):I = R(I)
7030  IF I = 0 THEN  GOTO 7100
7040  IF K < 18 THEN K = K + 1:
      GOTO 7020
7045  VTAB 20: PRINT "-->"
7050  GET I$: IF  ASC I$ = 21 THEN  HOME :
      GOTO 7020
7060  GOTO 1000
7100  PRINT "ENDE DER SCHLUESSELWORTE":
      GET A$: GOTO 1000
9000  VTAB 20: PRINT V$;" ?"
9005  GET I$: GOTO 1000
```

Abbildung 18: Programm Notizbuch.

Stichwort-Kartei 7

Dieses Programm entspricht in der äußeren Form dem vorangegangenen Programm. Es verwendet jedoch zum Aufbau der Kartei keine verketteten Listen, sondern binäre Bäume.

Eines der Hauptaufgabengebiete der Computer besteht in dem Abspeichern von Daten. Egal, ob es sich hierbei um Text oder Zahlen handelt, das Problem bleibt gleich: Wie speichert man die Daten ab, um sie bei Bedarf schnell wieder Finden, Ändern oder Löschen zu können.

Die Daten alphabetisch in ein Feld einzuspeichern geht nur im Speicher, nicht auf Diskette: Es müssten zu viele Einträge verschoben werden, falls etwas neu eingefügt oder gelöscht werden soll.

Man muss also dafür sorgen, daß beim Eintragen von neuen Daten möglichst wenig geändert werden muss. Eine der besten Methoden hierfür ist der Binärbaum: Man geht von einem Anfangselement aus, das einen Pointer zum nächsten Element besitzt. Um nun eine Ordnung zu erhalten, besitzt jedes Element nicht nur einen sondern 3 Pointer: Einen zum Vorgänger (nötig, falls etwas gelöscht werden soll), sowie zwei Pointer für den größeren bzw. kleineren Nachfolger (kleiner bedeutet hier - im Alphabet vorher, größer nachher kommend). (Siehe hierzu Abbildung 19).

Soll ein Element gesucht werden, so geht man vom Ausgangselement rechts bzw. links weiter, solange bis das Element gefunden ist, je nachdem, ob das Gefundene kleiner bzw. größer als das Gesuchte ist.

Wird ein neues Element eingetragen, so wird zuerst so getan, als ob man dieses Element suchen würde. Findet man

es tatsächlich, so ist es schon vorhanden (=>Errormeldung). Ansonsten kommt man irgendwann an das Ende eines Astes (ein Element, dessen Pointer in der gewünschten Richtung gleich 0 ist). Dieser Pointer wird jetzt auf das Ende der Diskettenaufzeichnung gesetzt und dort wird der neue Eintrag angefügt.

Zum Löschen ist etwas mehr Arbeit nötig:

Es gibt drei mögliche Fälle:

a) Das Element hat keinen Nachfolger; es muss also nur der Pointer des vorangegangenen Elements auf 0 gesetzt werden.

b) Das Element ist ein Nachfolger; die Pointer des Vorgängers werden auf den Nachfolger, die des Nachfolgers auf den Vorgänger des zu löschenden Elements gesetzt.

c) Das Element hat zwei Nachfolger; man muss das nächst kleinere Element finden und an den Platz des gelöschten Elements stellen (Pointer bleiben gleich). Man findet das nächst kleinere Element, indem man zuerst nach links, und dann soweit wie möglich nach rechts geht.

Jetzt muss nur noch der frei gewordene Platz gefüllt werden. Das geschieht, indem das letzte Element der Aufzeichnung dorthin verschoben wird, die Pointer zu und von diesem Element geändert werden, und die Längenangabe der Datei um 1 vermindert wird.

Doch nun zum Programm: Es gestattet die Eingabe von maximal 47 Zeichen Text pro Eintrag (Stichwort + Daten; 64 Zeichen Record-Länge minus 3 mal 4 Zeichen für den Pointer sowie 5 Zeichen für die Trennung der verschiedenen Einträge im Record).

Das Maschinensprache-Programm ist nur dazu da, auch die Eingabe von Kommas, Anführungszeichen und Doppelpunkten zu ermöglichen.

Wem die 47 Zeichen zu wenig sind, braucht nur in Zeile 3800 die 64 gegen die gewünschte Länge +17 (für die

Pointer), und die Abfrage in 1020 zu ändern.

Die Suchzeit beträgt nach eigenen Erfahrungen bei 650 Einträgen etwa 6 Sekunden, das ist relativ schnell.

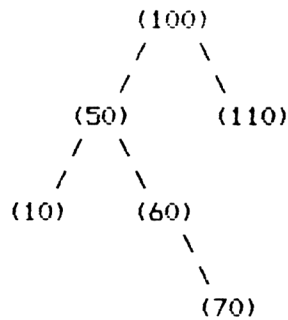


Abbildung 19: Aufbau eines binären Baumes.

```
 1  DATA 32,190,222,169,44,141,255,1,32,
    44,213,173,0,2,201,3,208,3,76,99,216,
    133,117,134,127,132,128,32,227,223,133,
    133,132,134,165,184,164,185,133,135,
    132,136,166,127,164,128,134,184,132,
    185,32,177
 2  DATA 0,165,184,164,185,32,88,3,32,
    61,231,32,123,218,32,183,0,165,184,164,
    185,133,127,132,128,165,135,164,136,
    133,184,132,185,96,169,0,133,14,133,
    13,76,237,227,255
 3  FOR I = 768 TO 863:
      READ A: POKE I,A:
    NEXT
10  REM
20  REM STICHWORT-KARTEI
30  REM
35  REM W.BAUER 1983
40  REM
90  D$ =  CHR$ (4)
```

```
 100  TEXT : HOME : PRINT "STICHWORT-KARTEI:
      ": PRINT "--------------------------------------":
      POKE 34,3
 110  INPUT "FILENAME:";F$:
      FL = 1: ONERR  GOTO 1900
 112  GOSUB 3800: PRINT D$"READ"F$",R0":
      INPUT FL
 115  POKE 216,0: PRINT D$"CLOSE"
 120  VTAB 1: HTAB 25: PRINT "MENUE
         ": HOME : PRINT  TAB( 5)"1:
       EINGABE": PRINT : PRINT  TAB( 5)"2:
       SUCHEN": PRINT : PRINT  TAB( 5)"3:
       ALPHABETISCHE AUSGABE":
      PRINT : PRINT  TAB( 5)"4:
       LOESCHEN"
 130  VTAB 18: INPUT "WELCHE FUNKTION:
      ";A$
 140  A =  VAL (A$): IF A < 1 OR A > 4 THEN 130
 150  ON A GOTO 1000,2000,3000,4000
1000  VTAB 1: HTAB 25: PRINT "EINGABE":
      HOME
1010  PRINT : PRINT "KEYWORD:
      ": CALL 768,KW$: IF KW$ = "" THEN 120
1020  PRINT : PRINT "DATEN:
      ": CALL 768,DA$: IF  LEN (DA$ + KW$)
      > 47 THEN  PRINT "ZU LANG !!":
      GOTO 1010
1030  IF DA$ = "" THEN 1010
1100  GOSUB 3800: IF FL = 1 THEN  PRINT D$"WRITE"F$",
      R1": PRINT KW$: PRINT DA$:
      GOTO 1800
1120  GOSUB 3900: IF KW$ = K0$ THEN  PRINT "SCHON
      VORHANDEN !!": GOTO 1820
1130  IF KW$ < K0$ THEN 1160
1140  IF R <  > 0 THEN T = R:
      GOTO 1120
1150  PRINT D$"WRITE"F$",R"T:
      PRINT K0$: PRINT D0$:
      PRINT B","L","FL: PRINT D$"WRITE"F$",
      R"FL: PRINT KW$: PRINT DA$:
      GOTO 1800
1160  IF L <  > 0 THEN T = L:
      GOTO 1120
```

```
1170  PRINT D$"WRITE"F$",R"T:
      PRINT K0$: PRINT D0$:
      PRINT B","FL","R: PRINT D$"WRITE"F$",
      R"FL: PRINT KW$: PRINT DA$:
      GOTO 1800
1800  PRINT T",0,0":FL = FL + 1:
      PRINT D$"WRITE"F$",R0":
      PRINT FL
1820  PRINT D$"CLOSE": GOTO 1010
1900  POKE 216,0: IF  PEEK (222) = 5 THEN 115
1910  PRINT : PRINT "ERROR:
      " PEEK (222): GET A$:
      GOTO 120
2000  VTAB 1: HTAB 25: PRINT "SUCHEN":
      HOME : PRINT "WILDCARD CHARACTER:
       '@'": PRINT : PRINT "BEISPIEL:
       'COM@' SUCHT ALLE WOERTER DIE MIT 'COM'
      BEGINNEN"
2010  PRINT  CHR$ (4)"CLOSE":
      PRINT : PRINT "KEYWORD:
      ": CALL 768,KW$: IF KW$ = "" THEN 120
2015  IF  RIGHT$ (KW$,1) = "@" THEN KW$
      =  LEFT$ (KW$, LEN (KW$) - 1):
      DA$ = KW$ +  CHR$ (127):
      GOSUB 3040: GOTO 2010
2020  GOSUB 3800
2040  GOSUB 3900: IF KW$ = K0$ THEN  PRINT ">>>"D0$:
      GOTO 2010
2050  IF KW$ > K0$ THEN 2080
2060  IF L <  > 0 THEN T = L:
      GOTO 2040
2070  PRINT "NICHT VORHANDEN !!":
      GOTO 2010
2080  IF R = 0 THEN 2070
2090  T = R: GOTO 2040
3000  VTAB 1: HTAB 25: PRINT "ALPHA OUT":
      HOME
3010  PRINT : PRINT "VON KEYWORD:
      ": CALL 768,KW$: IF KW$ = "" THEN 120
3020  PRINT : PRINT "BIS KEYWORD:
      ": CALL 768,DA$
3030  IF KW$ > DA$ THEN A$ = KW$:
      KW$ = DA$:DA$ = A$
```

```
3040  GOSUB 3800
3100  GOSUB 3900: IF KW$ = K0$ THEN 3420
3110  IF KW$ < K0$ THEN 3140
3120  IF R <  > 0 THEN T = R:
      GOTO 3100
3130  GOTO 3430
3140  IF L <  > 0 THEN T = L:
      GOTO 3100
3150  GOTO 3420
3400  GOSUB 3900
3410  IF L <  > 0 THEN T = L:
      GOTO 3400
3415  IF K0$ > DA$ THEN 3460
3420  PRINT K0$ TAB( 20)D0$
3430  IF R <  > 0 THEN T = R:
      GOTO 3400
3440  A = T:T = B: GOSUB 3900:
      IF A = L THEN 3415
3450  IF A <  > 1 THEN 3440
3460  PRINT D$"CLOSE": IF  RIGHT$ (DA$,1)
      =  CHR$ (127) THEN  RETURN
3470  GOTO 3010
3800  PRINT D$"OPEN"F$",L64":
      T = 1: RETURN
3900  PRINT D$"READ"F$",R"T:
      CALL 768,K0$: CALL 768,D0$:
      INPUT B,L,R: RETURN
4000  VTAB 1: HTAB 25: FLASH :
      PRINT "LOESCHEN": NORMAL :
      HOME
4010  PRINT : PRINT "KEYWORD:
      ": CALL 768,KW$: IF KW$ = "" THEN 120
4020  GOSUB 3800
4030  GOSUB 3900: IF KW$ = K0$ THEN 4100
4040  IF KW$ > K0$ THEN 4070
4050  IF L <  > 0 THEN T = L:
      GOTO 4030
4060  PRINT "NICHT VORHANDEN":
      GOTO 4290
4070  IF R <  > 0 THEN T = R:
      GOTO 4030
4080  GOTO 4060
4100  T2 = T: IF L <  > 0 THEN 4310
```

```
4110  IF R <  > 0 THEN T = R:
      GOTO 4300
4120  T = B:TX = 0: IF T2 = 1 THEN  PRINT "SPEICHER
      VOLLSTAENDIG GELOESCHT!!":
      PRINT D$"DELETE"F$:
      GET A$: GOTO 100
4130  GOSUB 3900: IF T2 = L THEN L = TX:
      GOTO 4150
4140  R = TX
4150  PRINT D$"WRITE"F$",R"T:
      PRINT K0$: PRINT D0$:
      PRINT B","L","R
4160  T = FL - 1: IF T2 = T THEN 4280
4165  GOSUB 3900: PRINT D$"WRITE"F$",R"T2:
      PRINT K0$: PRINT D0$:
      PRINT B","L","R
4170  LL = L:RL = R:T = B:
      GOSUB 3900: IF L = FL - 1 THEN L = T2:
      GOTO 4190
4180  R = T2
4190  PRINT D$"WRITE"F$",R"T:
      PRINT K0$: PRINT D0$:
      PRINT B","L","R
4200  IF RL <  > 0 THEN T = RL:
      GOSUB 3900: PRINT D$"WRITE"F$",R"T:
      PRINT K0$: PRINT D0$:
      PRINT T2","L","R
4210  IF LL <  > 0 THEN T = LL:
      GOSUB 3900: PRINT D$"WRITE"F$",R"T:
      PRINT K0$: PRINT D0$:
      PRINT T2","L","R
4280  FL = FL - 1: PRINT D$"WRITE"F$",R0":
      PRINT FL
4290  PRINT D$"CLOSE": GOTO 4010
4300  IF T2 = 1 THEN 4400
4305  BL = B: GOSUB 3900:
      PRINT D$"WRITE"F$",R"T:
      PRINT K0$: PRINT D0$:
      PRINT BL","L","R:TX = T:
      T = BL: GOTO 4130
4310  IF R = 0 THEN T = L:
      GOTO 4300
```

```
4320  T = L:LL = L:RL = R:
      BL = B
4330  GOSUB 3900: IF R <  > 0 THEN T = R:
      GOTO 4330
4340  PRINT D$"WRITE"F$",R"T2:
      PRINT K0$: PRINT D0$:
      PRINT BL","LL","RL:
      GOTO 4100
4400  GOSUB 3900: PRINT D$"WRITE"F$",R1":
      PRINT K0$: PRINT D0$:
      PRINT "1,"L","R:T2 = T
4410  LL = L: IF R <  > 0 THEN T = R:
      GOSUB 3900: PRINT D$"WRITE"F$",R"T:
      PRINT K0$: PRINT D0$:
      PRINT "1,"L","R
4420  IF LL <  > 0 THEN T = LL:
      GOSUB 3900: PRINT D$"WRITE"F$",R"T:
      PRINT K0$: PRINT D0$:
      PRINT "1,"L","R
4430  GOTO 4160
```

Abbildung 20: Programm KARTEI.

```
STICHWORT-KARTEI:        MENUE
----------------------------------------

    1: EINGABE

    2: SUCHEN

    3: ALPHABETISCHE AUSGABE

    4: LOESCHEN

WELCHE FUNKTION:


STICHWORT-KARTEI:        EINGABE
----------------------------------------


KEYWORD:
KARLSRUHE

DATEN:
AM 10.9.84 FIRMA SONNTAG BESUCHEN


STICHWORT-KARTEI:        SUCHEN
----------------------------------------

WILDCARD CHARACTER: '@'

BEISPIEL: 'COM@' SUCHT ALLE WOERTER DIE
MIT 'COM' BEGINNEN

KEYWORD:
K@
KARLSRUHE           AM 10.9.84 FIRMA SONN
TAG BESUCHEN

KEYWORD:
```

Abbildung 21: Probelauf Programm KARTEI.

Notizen

AVL-Bäume 8

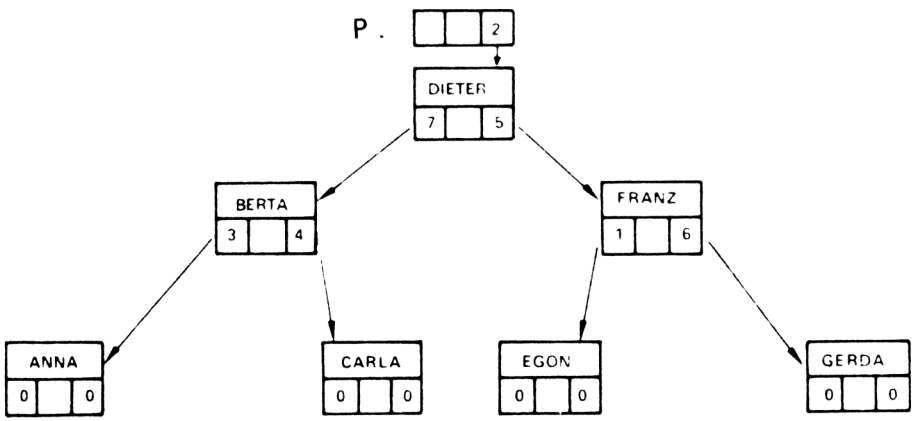


Abbildung 22: Binärer Baum.

1 EGON
2 DIETER
3 ANNA
4 CARLA
5 FRANZ
6 GERDA
7 BERTA

Abbildung 23: Namenliste.

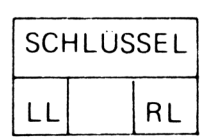


Abbildung 24: Knoten.

Die Abbildung 22 zeigt einen geordneten, ausbalancierten, binären Baum. Auf einer Diskette sind die Datensätze für die Vornamen in der Abbildung 23 gezeigten Reihenfolge gespeichert. Die Vornamen sind dabei die Schlüsselworte, nach denen der binäre Baum aufgebaut wurde. Jeder Knoten enthält neben dem Schlüsselwort noch zwei Zeiger RL und LL. Der Zeiger RL weist auf das nächst größere Schlüsselwort, das rechts vom betrachteten Knoten liegt. LL weist auf das nächst kleinere Element hin. Den Aufbau eines Knotens zeigt Abbildung 24.

Das oberste Element des Baumes ist das Schlüsselwort DIETER. Auf diesen Knoten zeigt ein Zeiger P, der den Stamm des Baumes markiert. P=2 zeigt, daß DIETER an zweiter Stelle auf der Diskette gespeichert ist. Der Zeiger RL=5 zeigt auf FRANZ, der Zeiger LL=7 auf BERTA.

Von BERTA wird auf CARLA (RL=4) und ANNA (LL=3) verwiesen. Beide Namen stellen das Ende der Verzweigungen dar. Die Zeiger RL und LL in beiden Knoten sind Null.

Wird nun ein Name gesucht, so wird der erste Vergleich mit P=2 begonnen und nach rechts oder links weitergegangen bis der gesuchte Name gefunden ist. Wird der Name EGON gesucht, so wird beim ersten Vergleich festgestellt, daß EGON größer (in alphabetischer Reihenfolge) als DIETER ist. Beim Vergleich mit FRANZ ist EGON kleiner und beim nächsten Vergleich ist EGON gefunden.

Der Zeiger LL, der auf EGON zeigt, ist 1. Somit ist EGON der erste Datendatz auf der Diskette. Ingesamt waren drei Vergleiche notwendig, bis der Name gefunden war. Bei 15 Namen sind es 4, bei 21 Namen 5 usw. Soll aus 4095 Datensätzen ein Name gesucht werden, so sind maximal 12 Vergleiche notwendig. Wird aber die Anzahl der Datensätze auf 8190 verdoppelt, so ist nur ein weiterer Vergleich (ingesamt dann 13) notwendig. Dies setzt aber voraus, daß der binäre Baum regulär und ausbalanciert ist. Auf diese Problematik soll gleich eingegangen werden.

Wo können solche Baumstrukturen verwendet werden?

Sie können überall dort verwendet werden, wo ein Schlüsselwort nicht zweimal vorkommen kann. Dies ist zum Beispiel bei Inventarverzeichnissen der Fall, wo eine Inventarnummer keine zwei verschiedenen Einträge bezeichnen kann. Für Bankleitzahlen oder für Versicherungsnummern kann diese Struktur ebenfalls verwendet werden. Auch Adressverwaltungen können aufgebaut werden. Der Herr Peter Fischer aus Rastatt wird von Herrn Peter Fischer aus Ulm dadurch unterschieden, daß seinem Namen ein beliebiges Zeichen angehängt wird, das beim Ausdruck von Adressen unterdrückt wird.

Von dieser Art von binären Bäumen muss man andere Arten von Bäumen unterscheiden, wie sie bei Suchbäumen oder Expertensystemen aufgebaut werden. Bei diesen wird in einem Knoten nach Eigenschaften verzweigt.

Zum Beispiel:

Bluttemperatur größer 37.8 C dann rechts weiter, weil krank, sonst links weiter, weil gesund.

Nun wieder zurück zu unserem binären Baum. In diesen soll ein neuer Name HILDE eingefügt werden. Der Platz dafür wird nach GERDA gefunden. Soweit ist noch alles in Ordnung. Der nächste Name KARIN würde seinen Platz nach Hilde erhalten. Unser schöner, binärer Baum gerät außer Kontrolle.

Der Aufbau eines Baumes ist also im wesentlichen durch die Reihenfolge der Eingabe der Schlüsselworte bestimmt. Wären die Namen in alphabetischer Reihenfolge eingegeben worden, so wäre ein Baum nach Abbildung 25 entstanden.

Soll hier ein Name gesucht werden, so müssen maximal sieben, statt drei Vergleiche durchgeführt werden. Nicht besser ist es mit einem Zick-Zack-Baum in Abbildung 26.

Solche Baumstrukturen sind selten und im allgemeinen ist ein binärer Baum bei einer zufälligen Eingabe von Schlüsselworten in den meisten Fällen einigermaßen ausbalan-

ciert. Neben anderen Verfahren um binäre Bäume auszubalancieren, haben zwei russische Mathematiker ANDELSON-VELSKII und LANDIS ein Verfahren angegeben, das einen binären Baum bei der Eingabe ausbalanciert hält. Das Verfahren für diese AVL-Bäume soll nun gezeigt werden.

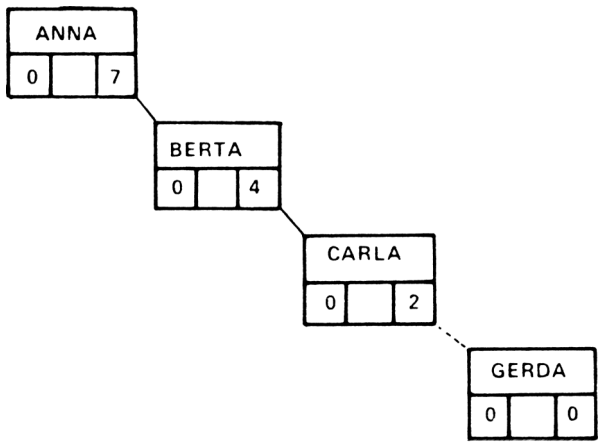


Abbildung 25: Entarteter Baum.

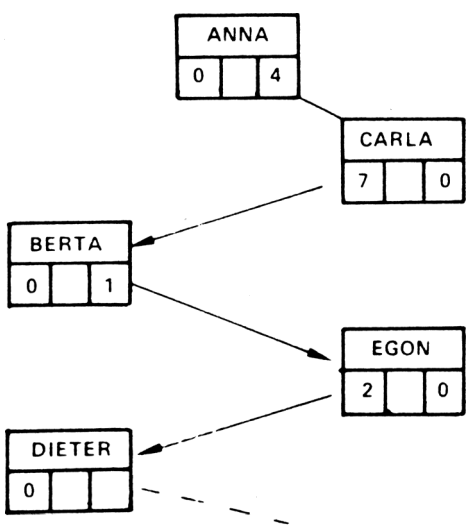


Abbildung 26: Zick-Zack Baum.

Zu den beiden Zeigern RL und LL wird eine weitere Information, der Faktor B hinzugefügt. Dieser Faktor gibt an, ob ein Baum nach rechts oder links wächst, oder ob er ausbalanciert ist.

B=1 bedeutet, daß der Baum nach rechts, B=-1, da er nach links um eine Höhenstufe gewachsen ist. B=0 bedeutet daß der Baum, oder ein Teil des Baumes, ab dieser Stelle nach rechts oder links gleich groß ist. B ist ebenfalls Null an den Enden des Baumes.

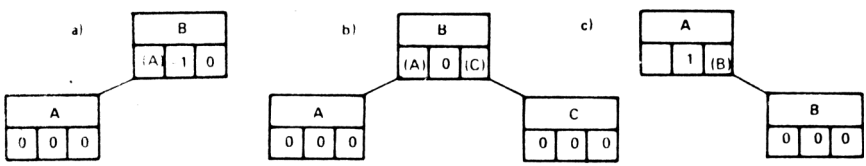


Abbildung 27: Die drei möglichen Fälle für den Faktor B.

In den folgenden Beispielen wird als Schlüsselwort ein Buchstabe verwendet. Betrachten wir nun Fall 27c. Dort soll ein neuer Knoten C angekündigt werden. Der Platz dafür wird rechts von B gefunden. Da der Baum nach rechts gewachsen ist, muss B(B)=1 werden, ebenso B(A). Dies ist ist aber schon Eins. Dadurch wird angezeigt, daß der Baum entartet und somit neu ausbalanciert werden muss.

In diesem Fall wird eine einzige Verschiebung durchgeführt. B wird der neue Stamm. Nach LL(B) wird die Adresse von A geschrieben, RL(A) wird Null und der Anfangszeiger P enthält die Adresse von B. Die Schreibweise <B> in der Zeichnung soll anzeigen, daß nicht das Schlüsselwort A, sondern die Adresse von A gespeichert ist. Die gleiche Vertauschung wird bei einem Baum angewendet, der zweimal nach links gewachsen ist. Hier sind dann die Zeiger RL und LL vertauscht. In beiden Fällen entsteht ein ausbalancierter Baum und alle Faktoren B sind Null.

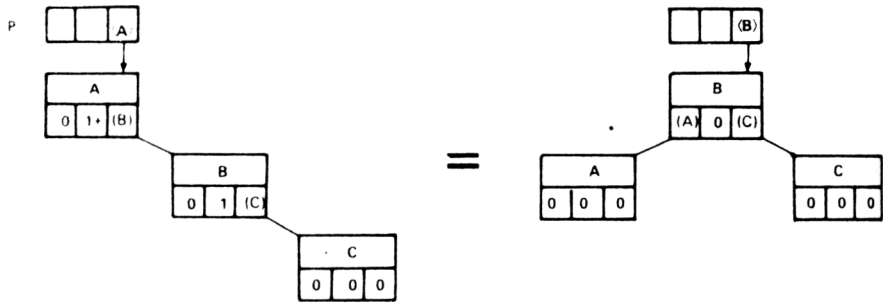


Abbildung 28: Balancieren eines entarteten Baumes. Einfache Verschiebung.

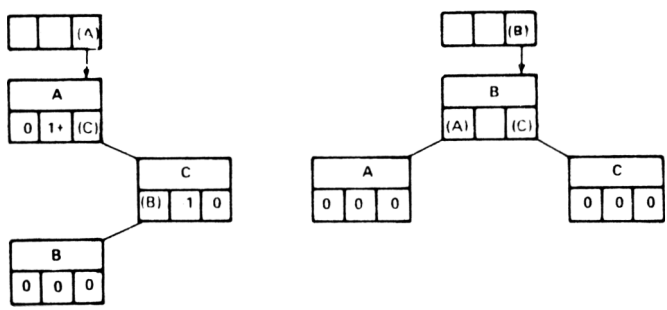


Abbildung 29: Balancieren eines entarteten Baumes. Zweifache Verschiebung.

Einen zweiten Fall zeigt Abbildung 29. Der Baum enthielt die Elemente A und C. Ein neues Element B wurde hinzugefügt. Von C aus gesehen, ist der Baum nach links gewachsen, B(C)=-1.

Von A aus aber nach rechts. Da aber B(A)=1 ist, muss neu ausbalanciert werden. In diesem Fall ist eine zweimalige Verschiebung notwendig. Da B>A, aber B<C ist, muss B zwischen A und C eingefügt und somit ein neuer Stamm werden. Die Zeiger der drei Elemente müssen entsprechend vertauscht werden. Entsprechendes gilt für einen Baum,

der nach links und dann nach rechts gewachsen ist.

Ein BASIC-Programm dazu zeigt Abbildung 30. Das Programm wurde nach Donald E. KNUTH, "The Art of Computer Programming, Vol.3, Searching and Sorting" programmiert. Anfängliche Schwierigkeiten beim Programmieren des auf Seite 455 bis 456 beschriebenen Algorithmus waren darauf zurückzuführen, daß die Angabe, der Baum sei nicht leer, (P)=0, zu ungenau ist. Dies setzt nur voraus, daß mindestens ein Element vorhanden ist.

Der angegebene Algorithmus funktioniert aber nur, wenn der Anfangsbaum ein "junger Baum" mit einem Stamm und zwei Ästen ist. Im Progrmm wird daher in den Zeilen 400 bis 590 dieser Baum aus drei Elementen aufgebaut. Das Einfügen des ersten und zweiten Elementes bereitet keine Schwierigkeiten. Beim dritten Element müssen alle vier Fälle für die Ausbalancierung programmiert werden. (Zeilen 485 bis 510).

Danach kann der Algorithmus angewendet werden. Er beginnt mit dem Aufsuchen des Platzes (Zeilen 200 bis 290). Ist AS, das neue Schlüsselwort, kleiner KEY$(P), so wird nach links weiter gegangen, anderenfalls nach rechts. Ist A$=KEY$(P), so ist das Schlüsselwort schon vorhanden und das Suchen ist beendet. Ein Platz ist gefunden, wenn Q=RL(P) oder Q=LL(P) ist. Bei dieser Suche werden zwei Zeiger T und S mitgeführt. S zeigt immer auf das letzte Element, bei dem B(K) (K für Knoten) ungleich Null ist. T ist jeweils der Vater (Vorgänger) von S.

Nachdem ein Platz gefunden ist, wird das neue Schlüsselwort im Unterprogramm 800 in den Baum eingefügt. Der nächste Schritt ist nun das Einfügen des Faktors B. Zwischen R=S(P) und dem neuen Element müssen alle B(K) +1 sein, wenn der Baum nach rechts, und -1, wenn der Baum nach links gewachsen ist. (Zeilen 600 bis 679). Nun muss festgestellt werden, ob neu ausbalanciert werden muss. Ist zum Beispiel B(S)=1 und der neue Faktor -1, so ist der Baum besser ausbalanciert, B(S) wird Null und der Algorithmus ist beendet. Ist aber B(S)=1 und der neue

Faktor ebenfalls 1, so muß ausbalanciert werden. Dies geschieht in den Zeilen 1000 bis 1330.

Hier sind die vier Fälle für rechts und links, einfaches und zweifaches Verschieben programmiert. Als letztes wird in den Zeilen 1400 bis 1430 der Zeiger P, im Programm RL(0), auf einen eventuellen neuen Stamm gesetzt.

Eine Bildschirmausgabe des jeweiligen Baumes ist ab Zeile 2000 programmiert. Dort wird die horizontale und vertikale Position für einzelne Buchstaben oder kurze Schlüsselworte berechnet. Nach jeder Eingabe wird der Baum auf den Bildschirm ausgegeben.

```
 50  N = 32
100  DIM KEY$(N),RL(N),LL(N),B(N)
105  GOSUB 2000
110  RL(0) = 0:LL(0) = 0:
     B(0) = 0:K = 1
115  HOME
120  IF K > 1 THEN  GOSUB 2100
122  VTAB 18: HTAB 1: INPUT "KEYWORD:
     ";A$
125  IF A$ = "" THEN  STOP
130  IF K < 4 GOTO 400
200  REM  SEARCH
210  P = RL(0):T = 0:S = P
220  IF A$ < KEY$(P) THEN Q = LL(P):
     BO =  - 1
230  IF A$ > KEY$(P) THEN Q = RL(P):
     BO = 1
240  IF A$ = KEY$(P) THEN  PRINT "SCHON
     IN LISTE": GOTO 120
250  IF Q = 0 AND BO =  - 1 THEN Q = K:
     LL(P) = Q: GOTO 290
260  IF Q = 0 AND BO = 1 THEN Q = K:
     RL(P) = Q: GOTO 290
270  IF B(Q) <  > 0 THEN T = P:
     S = Q
280  P = Q: GOTO 220
290  GOSUB 800: GOTO 600
400  REM  FIRST ELEMENT
```

```
405  IF RL(0) = 0 THEN RL(0) = K:
     Q = K: GOSUB 800: GOTO 120
407  IF K > 2 THEN  GOTO 440
410  REM  SECOND ELEMENT
415  P = RL(0)
420  IF A$ < KEY$(P) THEN LL(P) = K:
     B(P) =  - 1
425  IF A$ > KEY$(P) THEN RL(P) = K:
     B(P) = 1
430  IF A$ = KEY$(P) THEN  GOTO 240
435  Q = K: GOSUB 800: GOTO 120
440  REM  THIRD ELEMENT
445  P = RL(0)
450  IF (A$ < KEY$(P) AND LL(P) = 0) THEN LL(P)
     = K: GOTO 435
455  IF (A$ > KEY$(P) AND RL(P) = 0) THEN RL(P)
     = K: GOTO 435
460  IF A$ = KEY$(P) THEN  GOTO 240
465  IF A$ < KEY$(P) THEN R = LL(P):
     GOTO 475
470  R = RL(P)
475  IF A$ < KEY$(R) THEN BO =  - 1:
     LL(R) = K:B(R) = BO:
     Q = K: GOSUB 800: GOTO 485
480  IF A$ > KEY$(R) THEN BO = 1:
     RL(R) = K:B(R) = BO:
     Q = K: GOSUB 800
485  IF (B(P) = 1 AND B(R) = 1) THEN LL(R)
     = P:RL(P) = 0:
     GOTO 590
495  IF (B(P) =  - 1 AND B(R) =  - 1) THEN RL(R)
     = P:LL(P) = 0:
     GOTO 590
505  IF (B(P) = 1 AND B(R) =  - 1) THEN Q
     = LL(R):LL(Q) = P:
     RL(Q) = R:RL(P) = 0:
     LL(R) = 0:B(R) = 0:R = Q:
     GOTO 590
510  IF (B(P) =  - 1 AND B(R) = 1) THEN Q
     = RL(R):RL(Q) = P:
     LL(Q) = R:LL(P) = 0:
     RL(R) = 0:B(R) = 0:R = Q:
     GOTO 590
```

```
590  RL(0) = R:B(P) = 0:
     B(R) = 0: GOTO 120
600  REM  ADJUST
610  IF A$ < KEY$(S) THEN R = LL(S):
     BO =  - 1
620  IF A$ > KEY$(S) THEN R = RL(S):
     BO = 1
630  P = R
640  IF A$ < KEY$(P) THEN U = LL(P):
     BO =  - 1
650  IF A$ > KEY$(P) THEN U = RL(P):
     BO = 1
660  IF A$ = KEY$(P) THEN  GOTO 700
670  B(P) = BO:P = U: GOTO 640
700  REM  TEST FOR REBALANCING
705  IF A$ < KEY$(S) THEN BO =  - 1:
     GOTO 710
707  BO = 1
710  IF B(S) = 0 THEN B(S) = BO:
     LL(0) = LL(0) + 1: GOTO 120
715  IF B(S) =  - BO THEN B(S) = 0:
     GOTO 120
720  IF B(R) = BO THEN  GOTO 1000
730  IF B(R) =  - BO THEN  GOTO 1200
800  KEY$(Q) = A$:RL(Q) = 0:
     LL(Q) = 0:B(Q) = 0:K = K + 1:
     RETURN
900  FOR I = 0 TO K - 1
910    PRINT KEY$(I);" ";B(I);" ";LL(I);"
       ";RL(I)
920  NEXT I
930  RETURN
1000 REM  RIGHT BALANCING, SINGLE ROTATION
1010 IF BO =  - 1 THEN  GOTO 1100
1020 P = R:RL(S) = LL(R):
     LL(R) = S
1030 B(S) = 0:B(R) = 0:
     GOTO 1400
1100 REM  LEFT BALANCING, SINGLE ROTATION
1110 P = R:LL(S) = RL(R):
     RL(R) = S: GOTO 1030
1200 REM  RIGHT BALANCING, DOUBLE ROTATION
1210 IF BO =  - 1 THEN  GOTO 1300
```

```
1220  P = LL(R):LL(R) = RL(P):
      RL(P) = R
1230  RL(S) = LL(P):LL(P) = S
1240  IF B(P) = BO THEN B(S) =  - BO:
      B(R) = 0
1250  IF B(P) = 0 THEN B(S) = 0:
      B(R) = 0
1260  IF B(P) =  - BO THEN B(S) = 0:
      B(R) =  - BO
1265  B(P) = 0
1270  GOTO 1400
1300  REM  LEFT BALANCING,DOUBLE ROTATION
1310  P = RL(R):RL(R) = LL(P):
      LL(P) = R
1320  LL(S) = RL(P):RL(P) = S
1330  GOTO 1240
1400  REM  FINAL TOUCH
1420  IF S = RL(T) THEN RL(T) = P:
      GOTO 120
1430  LL(T) = P: GOTO 120
2000  REM  AUSGABE
2010  DIM PM(15,4),ZM(15,4)
2020  PM(0,0) = 20:PM(0,1) = 10:
      PM(1,1) = 30
2030  FOR I = 0 TO 3
2040    PM(I,2) = 5 + I * 10:
      NEXT I
2050  S = 1:C = 2
2060  FOR I = 0 TO 7:S =  - 1 * S:
        M =  INT (I / 2):PM(I,3) = PM(M,2) + S
        * C:
      NEXT I
2070  S = 1:C = 1
2080  FOR I = 0 TO 15:S =  - 1 * S:
        M =  INT (I / 2):PM(I,4) = PM(M,3) + S
        * C:
      NEXT I
2090  RETURN
2100  REM  BILDSCHIRMAUSGABE
2104  O = K - 1
2105  L =  INT ( LOG (O) /  LOG (2))
2107  HOME :S = 2
```

```
2110  ZM(0,0) = RL(0):P = RL(0)
2115  VTAB S: HTAB PM(0,0):
      PRINT KEY$(P)
2117  IF L = 0 THEN  RETURN
2122  VTAB S + 2:S = S + 2
2124  R = LL(P): IF R <  > 0 THEN  HTAB PM(0,
      1): PRINT KEY$(R);
2125  ZM(0,1) = R
2126  R = RL(P): IF R <  > 0 THEN  HTAB PM(1,
      1): PRINT KEY$(R)
2127  ZM(1,1) = R
2128  IF L = 1 THEN  RETURN
2130  J = 0
2135  FOR I = 0 TO 1
2140    ZM(J,2) = LL(ZM(I,1)):
        P = ZM(J,2): HTAB PM(J,2):
        IF P <  > 0 THEN  PRINT KEY$(P);
2145    J = J + 1
2150    ZM(J,2) = RL(ZM(I,1)):
        P = ZM(J,2): HTAB PM(J,2):
        IF P <  > 0 THEN  PRINT KEY$(P);
2155    J = J + 1
2160  NEXT I
2170  IF L = 2 THEN  RETURN
3000  FOR J = 0 TO 4
3010    FOR I = 0 TO 15
3020      PRINT ZM(I,J);
3030    NEXT I
3050    PRINT
3060  NEXT J
```

Abbildung 30: Programm AVL-Bäume.

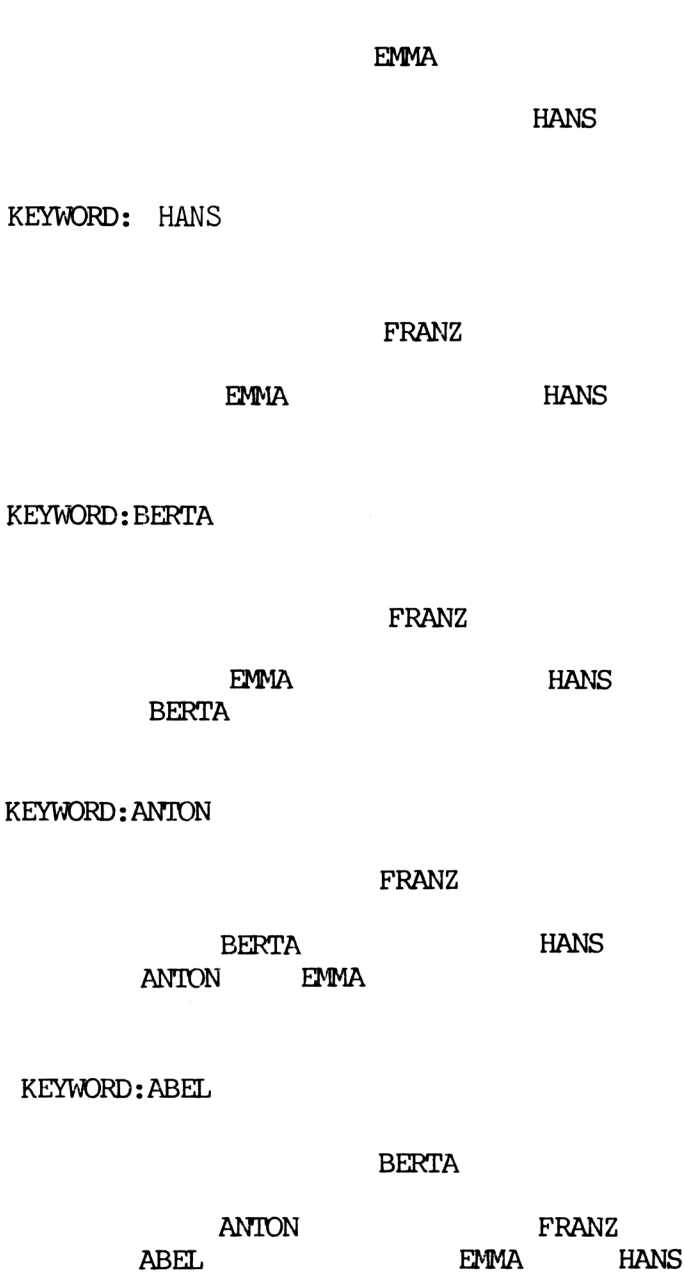


Abbildung 31: Ein Beispiel.

Notizen

Zeichnen eines Balken-diagramms

9

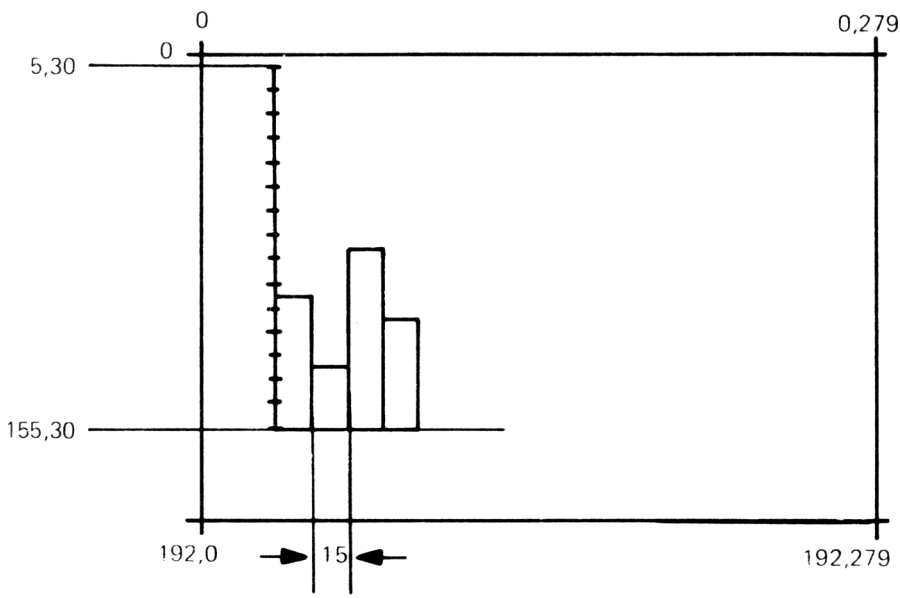


Abbildung 32: Koordinaten der Bildpunkte.

Die Abbildung zeigt die Koordinaten der Bildpunkte des APPLE II. Der linke Bildpunkt hat die Koordinaten x=0, y=0. Die Auflösung beträgt in HGR 279 Bildpunkte in x-Richtung und 192 Bildpunkte in y-Richtung. Die Anfangskoordinaten für die y-Achse sind x=30, y=155. Die Länge beträgt 150 Bildpunkte. Ein Balken ist 15 Bildpunkte breit. Die Beschriftung der Achse und der Datenelemente erfolgt durch SHAPES. In den Zeilen 30 bis

100 ist eine SHAPE-TABLE der Zeichen

- . / 0 1 2 3 4 5 6 7 8 9

angelegt. Der Befehl

DRAW 4 AT 140,90

zeichnet eine Null in die Mitte des Bildschirms. Die Parameter sind HCOLOR = 3:SCALE = 1 und ROT = 0.

```
 10  GOTO 1000
 30  HIMEM: 38144: LOMEM: 16384
 40  DATA  13,0,28,0,34,0,36,0,43,0,57,
     0,64,0,75,0,86,0,96,0,107,0,121,0,129,
     0
 50  DATA  141,0,192,56,13,45,4,0,4,0,216,
     12,12,12,12,4,0,101,36,36,28,63,23,54,
     54
 60  DATA  14,4,96,12,4,0,107,39,36,36,
     188,6,0,9,63,63,100,12,101,228,63,23
 70  DATA  6,0,216,14,45,12,228,103,12,
     60,63,55,0,33,44,28,36,188,23,23,46,
     45
 80  DATA  0,216,14,45,12,36,28,63,39,44,
     45,53,0,2,101,228,63,150,28,36,100
 90  DATA  12,45,6,155,0,35,36,97,12,60,
     63,55,0,101,228,103,33,28,63,23,174,
     23
100  DATA  118,5,0,216,42,101,12,36,36,
     59,63,50,14,45,5,0
110  FOR I = 0 TO 153
120    READ A: POKE 38144 + I,A
130  NEXT I
140  RESTORE
150  POKE 232,0: POKE 233,149
160  RETURN
400  REM  URSPRUENGLICHE DATEN
410  FOR I = 1 TO N
420    E(I) = S(I)
430  NEXT I:S = 0: RETURN
500  REM  BUBBLESORT
```

```
510  FOR I = 1 TO N:S(I) = E(I):
     NEXT I
520  FOR I = 1 TO N - 1
525    FOR J = I + 1 TO N
530      IF E(I) < E(J) THEN 540
535      H = E(I):E(I) = E(J):
         E(J) = H
540    NEXT J
550  NEXT I
560  S = 1: RETURN
600  HGR : HCOLOR= 3: SCALE= 1:
     ROT= 0
610  HPLOT 30,155 TO 30,5
620  FOR I = 0 TO 10
625    U = 155 - I * 15
630    HPLOT 28,U TO 32,U
640  NEXT I
650  FOR I = 0 TO 10
660    DRAW 4 AT 20,158 - I * 15
670  NEXT I
680  FOR I = 1 TO 10:A = I:
       IF I = 10 THEN A = 0
690    DRAW A + 4 AT 12,158 - I * 15
700  NEXT I
710  DRAW 5 AT 5,8
720  RETURN
1000 REM  DATA INPUT
1005 IF  PEEK (233) <  > 149 THEN  GOSUB 30
1007 S = 0
1010 TEXT : HOME : INPUT "ZAHL DER DATEN:
     ";N
1020 DIM E(N)
1030 FOR I = 1 TO N
1040   PRINT "X(";I;")=";:
       INPUT "";E(I)
1050 NEXT I
1055 IF S = 1 THEN  GOTO 1067
1057 GOSUB 1370: VTAB 23:
     HTAB 1
1060 PRINT "SOLLEN DATEN SORTIERT WERDEN
     (J/ ):": GET A$
1065 IF A$ = "J" THEN  GOSUB 500:
     GOTO 1085
```

```
1067  GOSUB 1370: VTAB 23:
      HTAB 1
1070  IF S = 1 THEN  PRINT "URSPRUENGLICHE
      DATEN (J/ ):";: GET A$
1075  IF A$ = "J" THEN  GOSUB 400
1085  GOSUB 1370: VTAB 23:
      HTAB 1: PRINT "100%=S)UMME,M)AXIMUM
      ODER F)AKTOR:"
1090  GET A$: IF A$ = "S" THEN  GOTO 1100
1095  IF A$ = "M" THEN  GOTO 1200
1097  IF A$ = "F" THEN  GOTO 1260
1099  GOTO 1090
1100  REM  SUM
1110  E = 0
1120  FOR I = 1 TO N:E = E + E(I):
      NEXT I
1130  FOR I = 1 TO N:P(I) = 155 - E(I) / E
        * 150:
      NEXT I
1140  GOTO 1300
1200  REM  MAXIMUM
1210  E = E(1)
1220  FOR I = 2 TO N
1230    IF E(I) > E THEN E = E(I)
1240  NEXT I
1250  GOTO 1130
1260  : REM  FAKTOR
1270  INPUT "F=";F
1275  FOR I = 1 TO N
1280    P(I) = 155 - F * E(I)
1290  NEXT I
1300  REM  PLOT DATA
1310  GOSUB 600
1315  HPLOT 30,155 TO 30 + 15 * N,155
1320  FOR I = 1 TO N
1325    U = 15 + 15 * I:
        V = 15 + U
1330    HPLOT U,155 TO U,P(I) TO V,P(I)
        TO V,155
1335  NEXT I
1340  FOR I = 1 TO N:A = I
1345    IF I > 9 THEN A = A - 9
1350    DRAW A + 3 AT 22 + 15 * I,150
```

```
1360  NEXT I
1365  GOSUB 1370: GOTO 1390
1370  VTAB 23: HTAB 1
1380  FOR I = 1 TO 35: PRINT " ";:
      NEXT I
1385  RETURN
1390  VTAB 23: HTAB 1: PRINT "NOCHMAL (
      /N):";: GET A$
1400  IF A$ = "N" THEN  TEXT :
     END
1410  GOSUB 1370: VTAB 23:
      HTAB 1:
1420  PRINT "NEUE DATEN (J/ ):
      ";: GET A$
1430  IF A$ = "J" THEN  GOTO 1000
1440  GOTO 1055
```

Abbildung 33: Programm: Zeichnen eines Balkendiagramms.

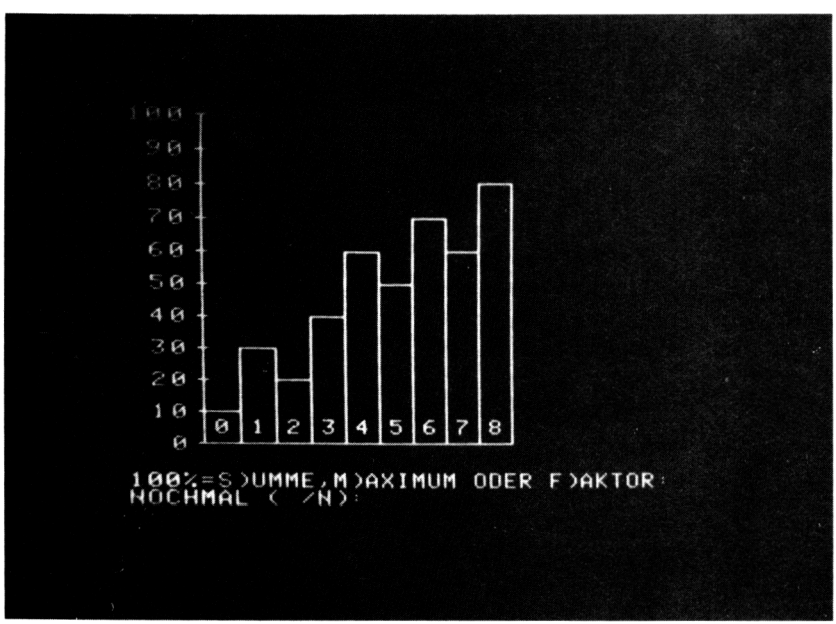


Abbildung 34: Beispiel Balkendiagramm.

Notizen

Zeichnen eines Tortendiagramms

10

Das Programm in Abbildung 35 zeichnet ein Tortendiagramm, wie es in Abbildung 36 gezeigt ist.

```
 10  GOTO 1000
 30  HIMEM: 38144: LOMEM: 16384
 40  DATA  13,0,28,0,34,0,36,0,43,0,57,
     0,64,0,75,0,86,0,96,0,107,0,121,0,129,
     0
 50  DATA  141,0,192,56,13,45,4,0,4,0,216,
     12,12,12,12,4,0,101,36,36,28,63,23,54,
     54
 60  DATA  14,4,96,12,4,0,107,39,36,36,
     188,6,0,9,63,63,100,12,101,228,63,23
 70  DATA  6,0,216,14,45,12,228,103,12,
     60,63,55,0,33,44,28,36,188,23,23,46,
     45
 80  DATA  0,216,14,45,12,36,28,63,39,44,
     45,53,0,2,101,228,63,150,28,36,100
 90  DATA  12,45,6,155,0,35,36,97,12,60,
     63,55,0,101,228,103,33,28,63,23,174,
     23
100  DATA  118,5,0,216,42,101,12,36,36,
     59,63,50,14,45,5,0
110  FOR I = 0 TO 153
120    READ A: POKE 38144 + I,A
130  NEXT I
140  RESTORE
370  POKE 232,0: POKE 233,149
375  RETURN
600  HGR : HCOLOR= 3: SCALE= 1:
     ROT= 0
610  PI = 3.151596235
```

```
 620  U0 = 140:V0 = 80:R = 70
 630  HPLOT U0 + R,V0
 640  FOR I = 0 TO 360 STEP 5
 650    W = I * PI / 180
 660    X = R *  COS (W):
        Y = R *  SIN (W)
 670    XP = X + U0:YP = V0 - Y
 675    HPLOT  TO XP,YP
 680  NEXT I
 690  RETURN
1000  REM  DATA INPUT
1005  IF  PEEK (233) <  > 149 THEN  GOSUB 30
1010  TEXT : HOME : INPUT "ZAHL DER DATEN
      ";N
1020  DIM E(N),F(N)
1030  FOR I = 1 TO N
1040    PRINT "(X";I;")=";:
        INPUT "";E(I)
1050  NEXT I
1060  FOR I = 1 TO 500:
      NEXT I
1100  REM  SUM
1110  E = 0
1120  FOR I = 1 TO N:E = E + E(I):
      NEXT I
1130  FOR I = 1 TO N:E(I) = E(I) / E * 360:
      NEXT I
1140  E = 0
1150  FOR I = 1 TO N
1160    LET E = E(I) + E:
        F(I) = E
1170  NEXT I
1300  REM  PLOT DATA
1310  GOSUB 600
1320  HPLOT U0,V0 TO U0 + R,V0
1325  FOR I = 1 TO N
1330    X = R *  COS (PI * F(I) / 180)
1340    Y = R *  SIN (PI * F(I) / 180)
1350    XP = U0 + X:YP = V0 - Y
1360    HPLOT XP,YP TO U0,V0
1370  NEXT I
1380  FOR I = 1 TO N
```

```
1390    W = PI * ((F(I) - F(I - 1)) / 2
        + F(I - 1)) / 180
1395    X = 2 * R / 3 *  COS (W):
        Y = 2 * R / 3 *  SIN (W)
1400    XP = X + U0:YP = V0 - Y
1410    N =  INT (E(I) / 3.6 + 0.5):
        NN =  INT (N / 10)
1420    DRAW NN + 4 AT XP,YP
1430    NN = N - 10 * NN
1440    DRAW NN + 4 AT XP + 7,YP
1450  NEXT I
1460  VTAB 24: PRINT
```

Abbildung 36: Beispiel Tortendiagramm.

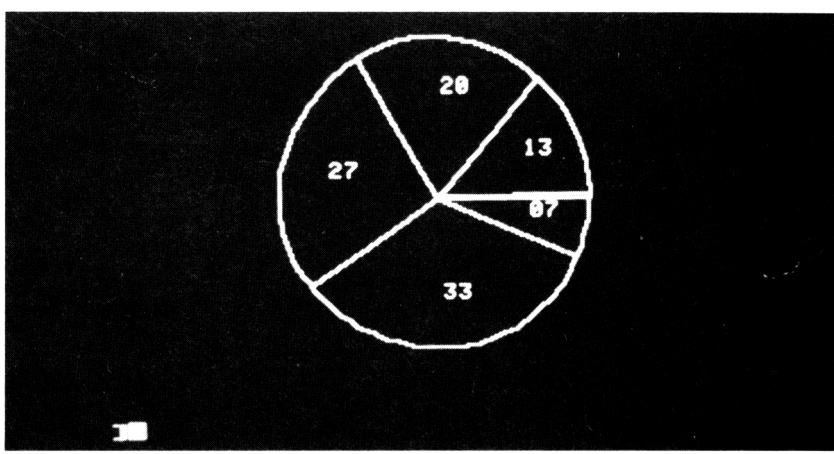


Abbildung 35: Programm: Zeichnen eines Tortendiagramms.

Notizen

Darstellung ebener Funktionen

11

Das Programm in Abbildung 37 dient zur Darstellung ebener Funktionen der Form y=f(x). Für diese Funktionen wird zunächst ein Achsenkreuz festgelegt. Ein Beispiel zeigt Abbildung 36.

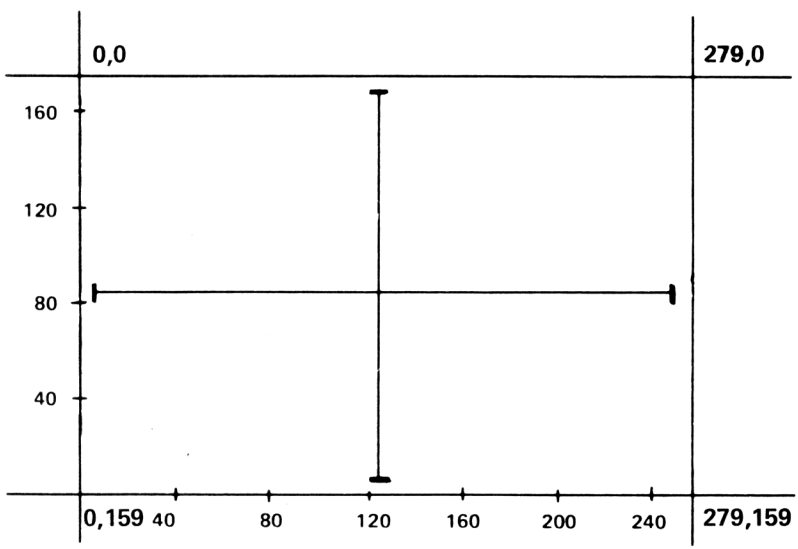


Abbildung 36: Festlegung eines Achsenkreuzes.

Die erste Eingabe ist der Nullpunkt U0, V0. Dieser Nullpunkt soll in der Mitte des Bildschirms liegen. Damit wird U0=140 und V0=80. Als nächstes werden die Achslängen eingegeben. Die x-Achse soll 200 Bildpunkte, die y-Achse soll 100 Bildpunkte lang sein. Damit ist XL=200 und YL=100.

Als Beispiel soll die Spaltfunktion y=sin(x)/x gezeichnet werden. Der Wertebereich für die x-Achse soll von XA=-10 bis XE=10 reichen. Für den Wertebereich der y-Achse müssen Abschätzungen gemacht werden. Für die Spaltfunktion ist YA=-0.5 und YE=1. In das Achsenkreuz werden Teilstriche eingezeichnet. Die x-Achse soll in 10 Teile, die y-Achse in 3 Teile eingeteilt werden. Damit wird XT=10 und YT=3.

Das Programm berechnet nach diesen Angaben die Maßstabsfaktoren und zeichnet das Achsenkreuz. Die Funktion ist in den Zeilen 1000 bis 1050 programmiert. Die Übergabeparameter an das Plotprogramm in den Zeilen 2400 bis 2490 sind die Variablen X und Y. Ein Punkt, der außerhalb des Bildschirms fällt, wird nicht gezeichnet.

Um nicht die Daten für die Zeichnung jedesmal neu einzugeben, werden diese in DATA-Anweisungen gespeichert und durch Unterprogramm ab Zeile 3000 eingelesen. Für die Eingabe der Werte muß Zeile 100 in

```
100 GOSUB 2000
```

geändert werden.

```
 100  GOSUB 3000: GOSUB 2200
1000  REM  FUNCTION
1005  VTAB 20: PRINT
1010  INPUT "SCHRITTWEITE S:
      ";S
1020  FOR X = XA TO XE STEP S
1030    IF X <  > 0 THEN Y =  SIN (X) / X
1040    GOSUB 2200
1050  NEXT X
1999  STOP
2000  REM   KOORDINATENSYSTEM
2005  HOME
2010  PRINT " DATEN FUER DAS KOORDINATENSYSTEM":
      PRINT
2015  INPUT "MITTELPUNKT DER KOORDINATEN
      X0,Y0:";X0,Y0
2020  INPUT "LAENGE DER ACHSEN XL,YL:
      ";XL,YL
```

```
2030  INPUT "GRENZEN VON X; XA,XE:
      ";XA,XE
2040  INPUT "GRENZEN VON Y; YA,YE:
      ";YA,YE
2045  INPUT "EINTEILUNG DER ACHSEN XT,YT:
      ";XT,YT
2050  XG =  ABS (XA) +  ABS (XE):
      YG =  ABS (YA) +  ABS (YE)
2060  Xl = XL / XG * XA:
      X2 = XL / XG * XE
2065  Xl = Xl + X0:X2 = X2 + X0
2068  YL =  - YL
2070  Yl = YL / YG * YA:
      Y2 = YL / YG * YE
2075  Yl = Yl + Y0:Y2 = Y2 + Y0
2080  MX = XL / XG:MY = YL / YG
2085  XT = XL / XT:YT = YL / YT
2100  HGR : HCOLOR= 3
2110  HPLOT Xl,Y0 TO X2,Y0
2120  HPLOT X0,Yl TO X0,Y2
2130  FOR X = Xl TO X2 STEP XT
2132    HPLOT X,Y0 - 2 TO X,Y0 + 2:
      NEXT X
2134  FOR Y = Yl TO Y2 STEP YT
2136    HPLOT X0 - 2,Y TO X0 + 2,Y:
      NEXT Y
2140  RETURN
2200  XP = X * MX + X0:YP = Y * MY + Y0
2205  HCOLOR= 3
2210  IF XP < 0 OR XP > 279 THEN  GOTO 2230
2220  IF YP < 0 OR YP > 159 THEN  GOTO 2230
2225  HPLOT XP,YP
2230  RETURN
3000  READ X0,Y0,XL,YL,XA,XE,YA,YE,XT,YT
3010  GOSUB 2050: RETURN
3020 END
3100  DATA    140,80,200, 100,-10,10,-0.5,
      1,10,3
```

Abbildung 37: Programm: Zeichnen von ebenen Funktionen.

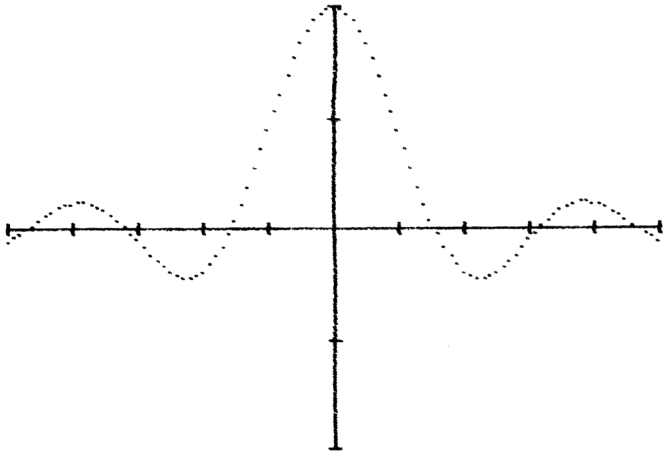


Abbildung 38: Spaltfunktion. y=sin(x)/x

Darstellung räumlicher Funktionen

12

Das Programm in Abbildung 41 stellt eine Erweiterung des vorherigen Programms zur Darstellung von ebenen Funktionen dar.

Mit dieser Erweiterung können 3-dimensionale Funktionen der Form y=f(x,z) in einer 2-dimensionalen Ebene dargestellt werden. Den Übergang vom 3-dimensionalen Koordinatensystem x,y,z in das 2-dimensionale Koordinatensystem zeigt Abbildung 40.

Der Winkel α ist der Winkel zwischen x- und y-Achse in der 2-dimensionalen Darstellung. Für die neuen Koordinaten u,v erhält man

$$u = x + x' \;,\; v = z + z' \quad \text{mit}$$

$$x' = y * \cos(\alpha) \quad \text{und}$$

$$z' = y * \sin(\alpha) \;.$$

Wie bei der Darstellung ebener Funktionen, werden die Daten für das Achsenkreuz in den Zeilen 2000 bis 2099 eingegeben. Die Koordinaten U0 und V0 beziehen sich auf den Nullpunkt 0,0 des Grafiksystems des APPLE. Dieser liegt in der linken oberen Ecke des Bildschirms.

Nach der Eingabe der Daten wird das Achsenkreuz im Unterprogramm ab Zeile 2100 gezeichnet. Auch hier können die Daten in DATA-Anweisungen im Programm gespeichert werden.

Die zu zeichnende Funktion ist in den Zeilen 1000 bis 1999 programmiert. Die Werte X,Y,Z werden and das Zei-

chen in den Zeilen 2400 bis 2460 übergeben. Sie werden mit den Maßstabsfaktoren multipliziert, in die Bildpunkte U,V umgerechnet und auf den Bildschirm ausgegeben. Punkte, die außerhalb des Bildschirms fallen, werden nicht gezeichnet.

Ein Beispiel zeigt Abbildung 42. Es werden drei Kreise in den Enenen x=const, y=const und z=const gezeichnet.

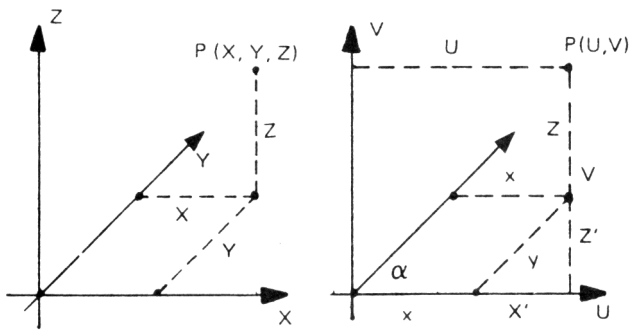


Abbildung 40: Darstellung einer räumlichen Funktion in einer Ebene.

```
 100  PI = 3.141592653
 110  GOSUB 3000
1000  REM  FUNKTION
1005  Z = 0:R = 0.75:S = 0.03
1010  FOR A = 0 TO 2 * PI STEP S
1020    X = R *  COS (A):
        Y = R *  SIN (A)
1030    GOSUB 2400:
      NEXT A
1040  Y = 0
1050  FOR A = 0 TO 2 * PI STEP S
1060    X = R *  COS (A):
        Z = R *  SIN (A)
1070    GOSUB 2400:
      NEXT A
```

```
1080  X = 0
1090  FOR A = 0 TO 2 * PI STEP S
1100    Y = R *  COS (A):
        Z = R *  SIN (A)
1110    GOSUB 2400:
      NEXT A
1999  STOP
2000  REM  KOORDINATENSYSTEM
2005  HOME
2010  PRINT " DATEN FUER DAS KOORDINATENSYSTEM":
      PRINT
2015  INPUT " MITTELPUNKT DER KOORDINATEN
      U0,V0:";U0,V0
2020  INPUT " LAENGE DER ACHSEN XL,YL,ZL:
      ";XL,YL,ZL
2025  INPUT " BEREICH VON X; XA,XE:
      ";XA,XE
2030  INPUT " BEREICH VON Y; YA,YE:
      ";YA,YE
2035  INPUT " BEREICH VON Z; ZA,ZE:
      ";ZA,ZE
2040  INPUT " EINTEILUNG DER ACHSEN XT,YT,
      ZT:";XT,YT,ZT
2045  INPUT " WINKEL ALPHA:
      ";WI
2050  XG =  ABS (XA) +  ABS (XE):
      YG =  ABS (YA) +  ABS (YE):
      ZG =  ABS (ZA) +  ABS (ZE)
2060  X0 = U0:Z0 = V0:Y0 = 0
2070  X1 = XL / XG * XA + X0:
      X2 = XL / XG * XE + X0
2080  Y1 = YL / YG * YA + Y0:
      Y2 = YL / YG * YE + Y0
2090  Z1 = ZL / ZG * ZA + Z0:
      Z2 = ZL / ZG * ZE + Z0
2100  MX = XL / XG:MY = YL / YG:
      MZ = ZL / ZG
2110  XT = XL / XT:YT = YL / YT:
      ZT = ZL / ZT
2120  CO =  COS (PI * WI / 180):
      SI =  SIN (PI * WI / 180)
2200  REM   KOORDINATENSYSTEM ZEICHNEN
```

```
2210  HGR : HCOLOR= 3
2220  HPLOT X1,Z0 TO X2,Z0
2230  HPLOT X0,Z1 TO X0,Z2
2240  FOR X = X1 TO X2 STEP XT
2250    HPLOT X,Z0 - 2 TO X,Z0 + 2:
      NEXT X
2260  FOR Z = Z1 TO Z2 STEP ZT
2270    HPLOT X0 - 2,Z TO X0 + 2,Z:
      NEXT Z
2280  UA = U0 + CO * Y1:
      VA = V0 - SI * Y1
2290  UE = U0 + CO * Y2:
      VE = V0 - SI * Y2
2300  HPLOT UA,VA TO UE,VE
2310  FOR Y = Y1 TO Y2 STEP YT
2320    U = U0 - CO * Y:
        V = V0 + SI * Y
2340    HPLOT U,V - 2 TO U,V + 2:
      NEXT Y
2350  RETURN
2400  REM   FUNKTION ZEICHNEN
2410  XP = X * MX:YP = Y * MX:
      ZP = Z * MZ
2420  U = XP + CO * YP + U0:
      V =  - ZP - SI * YP + V0
2430  IF U < 0 OR U > 279 THEN  GOTO 2460
2440  IF V < 0 OR V > 159 THEN  GOTO 2460
2450  HPLOT U,V
2460  RETURN
3000  READ U0,V0,XL,YL,ZL,XA,XE,YA,YE,ZA,
      ZE,XT,YT,ZT,WI
3010  GOTO 2050
3100  DATA    140,80,140,140,140,-1,1,-1,
      1,-1,1,4,4 ,4,45
```

Abbildung 41: Programm: 3D-PLOT.

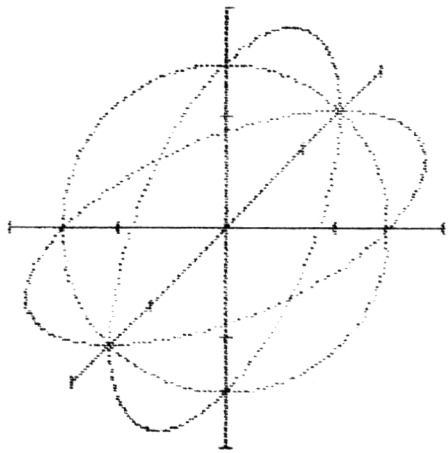


Abbildung 42: Beispiel 3D-PLOT.

Notizen

Computer-Grafiken 13

Die folgenden 4 Programme benutzen hochauflösende Grafik. Das erste zeichnet Umrisse, das zweite ist ein kleines Spiel und das dritte zeichnet die Umrisse des Mondes. Das vierte Programm erzeugt Unordnung.

13.1 POLYGON-GRAPHIK.

Dieses Programm zeichnet ein reguläres Vieleck um den Bildschirmmittelpunkt. Dieses Polygon wird um den Mittelpunkt und um eine Ecke gedreht. Um die Zeichengeschwindigkeit zu erhöhen sind der Sinus und der Cosinus in einer Tabelle gespeichert. Zeile 200 setzt den High Resolution Mode und HCOLOR auf Weiß. In Zeile 350 werden die Koordinaten des Bildschirmmittelpunktes bestimmt.

Zeile 530 zeichnet eine Linie von letzten Punkt zum Punkt X1 und Y1. Wenn das Bild fertig ist, piept das Programm und wartet auf einen Tastendruck. Abbildung 43 zeigt ein Beispiel.

Dies sind die Daten für dieses Bild:

```
DREHUNG1   = 44
DREHUNG2   = 25
GRÖSSE1    = 36
GRÖSSE2    = 35
ECKZAHL    =  4
GEZ.ECKEN  =  2
```

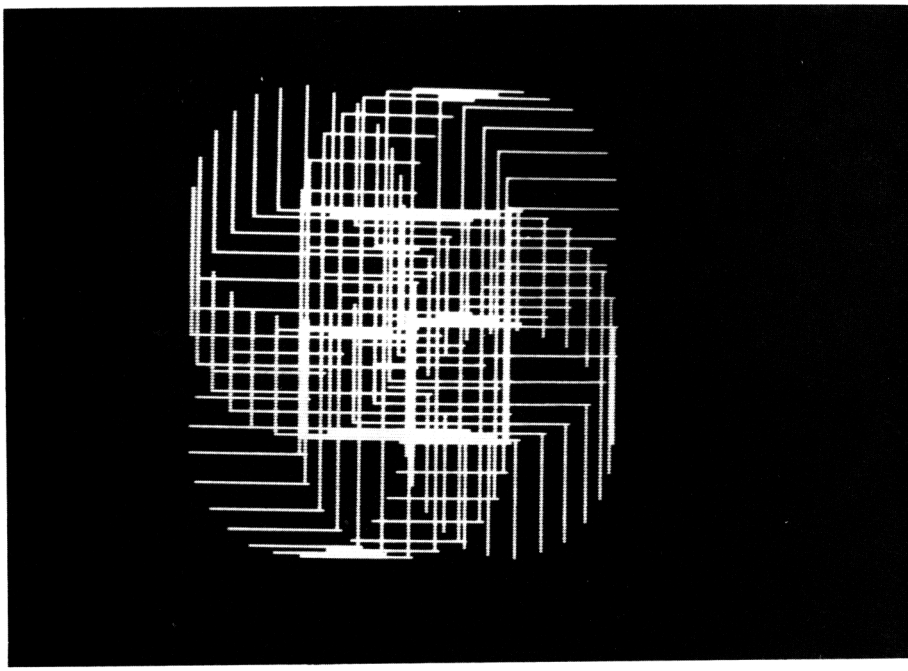


Abbildung 43: Beispiel Polygon-Graphik.

```
 10  DIM C(100),S(100):
     P = 3.14159265
 20  HOME : PRINT "COMPUTERGRAFIK":
     PRINT : PRINT "BITTE EINEN MOMENT WARTEN"
100  FOR I = 0 TO 2 * P STEP P / 50:
       S(X) =  SIN (I):C(X) =  COS (I):
       X = X + 1:
     NEXT
120  PRINT : INPUT "DREHUNG1:
     ";D1
130  INPUT "DREHUNG2:";D2
140  INPUT "GROESSE1:";G1
150  INPUT "GROESSE2:";G2
160  INPUT "ECKZAHL:";E:
     D3 = 100 / E
```

```
170  INPUT "GEZ.ECK:";E
200  HGR2 : HCOLOR= 3
300  R1 = 0:R2 = 0
350  X = 140:Y = 95: GOTO 410
400  X = 140:Y = 95
405  IF R1 < .1 OR R1 > 99.9 THEN  IF R2
     < .1 OR R2 > 99.9 THEN  PRINT
     CHR$ (7): GET A$:
     TEXT : GOTO 120
410  R1 = R1 + D1: IF R1 > 100 THEN R1
     = R1 - 100
420  X1 = X + C(R1) * G1:
     Y1 = Y + S(R1) * G1
430  HPLOT X1 + .5,Y1 + .5
500  FOR I = 1 TO E
510    R2 = R2 + D3: IF R2 > 100 THEN R2
       = R2 - 100
520    X1 = X1 + C(R2) * G2:
       Y1 = Y1 + S(R2) * G2
530    HPLOT  TO X1 + .5,Y1 + .5
     NEXT
540  R2 = R2 + D2: IF R2 > 100 THEN R2
     = R2 - 100
600  GOTO 400
```

Abbildung 44: Programm Polygon-Graphik.

Die Eingabewerte haben folgende Bedeutung:

DREHUNG1 : Rotation nach jeder Sequenz um die Bildschirmmitte. Dies ist eine Zahl von 0 bis 100, wobei 100 eine Drehung um 360 Grad bedeutet.
DREHUNG2 : Rotation des Polygons um eine Ecke.
GRÖSSE1 : Radius des Polygons.
GRÖSSE2 : Grösse des Polygons.
ECKZAHL : Anzahl der Ecken.
GEZ.ECKEN: Anzahl der gezeichneten Ecken.

Empfohlene Beispiele :

D1	D2	G1	G2	EZ	GEZ
44	25	36	35	4	2
2	99.5	45	45	1	1
2	99	40	40	3	3
46	12.5	10	35	8	4
12.5	42	50	20	4	4
40	42	50	20	5	5
98	3.2	50	30	1	1

13.2 IRRGARTEN.

Es gibt nur wenige Menschen die wirklich einmal in einem Irrgarten waren. Das Irrgartenprogramm erlaubt es durch einen Irrgarten zu gehen und zu versuchen den Ausgang zu finden. Zusätzkich kann man den Irrgarten von oben betrachten, um festzustellen wo man sich befindet. Die Größe des Irrgartens ist auf 38 in X-richtung und 22 in Y-Richtung begrenzt. VIEL GLUCK.

```
10  TEXT : NORMAL : HOME :
    PRINT  TAB( 11)"*** IRRGARTEN ***"
20  PRINT : PRINT : PRINT "ZIEL DIESES
    SPIELES IST ES, DEN AUSGANG"
30  PRINT : PRINT "EINES IRRGARTENS ZU
    FINDEN, IN DEM SIE"
40  PRINT : PRINT "SICH BEFINDEN !"
50  PRINT : PRINT "WENN SIE DIE TASTE
    'RETURN' DRUECKEN"
60  PRINT : PRINT "BEWEGEN SIE SICH EIN
    SEGMENT NACH VORNE,"
70  PRINT "WENN DIE TASTEN '<-' BZW.'->'
    GEDRUECKT"
80  PRINT : PRINT "WERDEN DREHEN SIE SICH
    UM 90 GRAD NACH"
81  PRINT : PRINT "LINKS BZW. NACH RECHTS."
82  PRINT : PRINT "DER AUSGANG IST DURCH
    EIN KREUZ GE-"
```

```
 83  PRINT : PRINT "KENNZEICHNET."
 84  PRINT : PRINT "BITTE TASTE DRUECKEN
     ->";: GET A$
 85  HOME : PRINT "WIRD 'X' GEDRUECKT ERSCHEINT
     DER IRR-": PRINT : PRINT "GARTEN VON
     OBEN."
 92  PRINT : PRINT : PRINT "EINGABE:
     ": PRINT
100  INPUT "GROESSE IN X-RICHTUNG:
     ";WX: PRINT : IF WX < 4 OR WX > 38 THEN
     PRINT "ZWISCHEN 4 UND 38 BITTE
     !!": PRINT : GOTO 100
105  INPUT "GROESSE IN Y-RICHTUNG:
     ";WY: PRINT : IF WY < 4 OR WY > 22 THEN
     PRINT "ZWISCHEN 4 UND 22 BITTE
     !!": PRINT : GOTO 105
106  IF  INT (WX / 2) <  > WX / 2 THEN WX
     = WX + 1
107  IF  INT (WY / 2) <  > WY / 2 THEN WY
     = WY + 1
110  PRINT : INPUT "SCHWIERIGKEITSSTUFE
     (1-4):";S: IF S < 1 OR S > 4 THEN 110
210  DIM A(WX,WY)
220  PRINT : PRINT "BITTE WARTEN":
     X =  INT (WX / 2):Y =  INT (WY / 2):
     A(X,Y) = 1
230  IF  RND (1) > .51 THEN XS =  SGN (
     RND (1) - .5):
     YS = 0: GOTO 250
240  XS = 0:YS =  SGN ( RND (1) - .5)
250  FOR I = 0 TO 1:X = X + XS:
       Y = Y + YS:A(X,Y) = 1
255    IF F THEN  IF X = 1 OR X = WX - 1
       OR Y = 1 OR Y = WY - 1 THEN 280
260    IF X = 0 OR X = WX OR Y = 0 OR Y
       = WY THEN 280
270  NEXT : GOTO 230
280  F = F + 1: IF F < S THEN 220
2000 HOME : HCOLOR= 3
2010 X =  INT (WX / 2):
     Y =  INT (WY) / 2:X1 = X:
     Y1 = Y:R = 0
```

```
2020  X(0) = 1:Y(0) = 0:
      X(1) = 0:Y(1) =  - 1:
      X(2) =  - 1:Y(2) = 0:
      X(3) = 0:Y(3) = 1
2030  RR = R + 1: IF RR > 3 THEN RR = 0
2035  XR = X(RR):YR = Y(RR)
2040  X1 = X:Y1 = Y:I = 1.5:
      HGR2 : GOSUB 3000
2050  I = I + .5:X1 = X1 + X(R):
      Y1 = Y1 + Y(R)
2051  IF X1 < 0 OR X1 > WX OR Y1 < 0 OR Y1
      > WY THEN  GOSUB 3300:
      GOTO 2065
2052  IF A(X1,Y1) = 0 THEN 2065
2055  GOSUB 3000
2060  GOTO 2050
2065  GOSUB 4000
2070  GET A$: IF A$ =  CHR$ (21) THEN R
      = R + 1: GOTO 2200
2080  IF A$ =  CHR$ (8) THEN R = R - 1:
      GOTO 2200
2085  IF A$ = "X" THEN  GOSUB 5000:
      GOTO 2040
2090  IF A$ <  >  CHR$ (13) THEN 2070
2100  X1 = X + X(R):Y1 = Y + Y(R):
      IF X1 < 0 OR X1 > WX OR Y1 < 0 OR Y1
      > WY THEN 6000
2110  IF A(X1,Y1) = 0 THEN 2070
2115  X = X1:Y = Y1: GOTO 2030
2200  IF R < 0 THEN R = 3:
      GOTO 2030
2210  IF R > 3 THEN R = 0
2220  GOTO 2030
3000  A1 = 140 + 139 / I:
      B1 = 95 + 94 / I:B2 = 191 - B1:
      A3 = 279 - A1:B3 = B1:
      B4 = B2
3010  C1 = 140 + 139 / (I - .5):
      D1 = 95 + 94 / (I - .5):
      D2 = 191 - D1:C3 = 279 - C1:
      D3 = D1:D4 = D2
3015  XT = X1 + XR:YT = Y1 + YR:
      IF XT < 0 OR XT > WX OR YT < 0 OR YT
      > WY THEN 3040
```

```
3020  IF A(XT,YT) = 1 THEN 3040
3030  HPLOT C1,D1 TO A1,B1:
      HPLOT C1,D2 TO A1,B2:
      GOTO 3050
3040  HPLOT C1,B1 TO A1,B1:
      HPLOT C1,B2 TO A1,B2
3050  HPLOT A1,B1 TO A1,B2
3090  XT = X1 - XR:YT = Y1 - YR:
      IF XT < 0 OR XT > WX OR YT < 0 OR YT
      > WY THEN 3140
3100  IF A(XT,YT) = 1 THEN 3140
3130  HPLOT C3,D3 TO A3,B3:
      HPLOT C3,D4 TO A3,B4:
      GOTO 3150
3140  HPLOT C3,B3 TO A3,B3:
      HPLOT C3,B4 TO A3,B4
3150  HPLOT A3,B3 TO A3,B4
3200  RETURN
3300  HPLOT A1,95 TO A3,95:
      HPLOT 140,B2 TO 140,B1:
      RETURN
4000  HPLOT A1,B1 TO A3,B1:
      HPLOT A1,B2 TO A3,B2:
      RETURN
5000  TEXT : HOME :
      FOR Z = 0 TO WY:
        FOR Q = WX TO 0 STEP  - 1
5010      IF A(Q,Z) = 0 THEN 5040
5015      INVERSE : IF Z = 0 OR Z = WY OR Q
          = 0 OR Q = WX THEN  FLASH :
          PRINT "A";: GOTO 5050
5020      IF Q = X THEN  IF Z = Y THEN
          FLASH : PRINT "*";:
          GOTO 5050
5040      PRINT " ";: NORMAL
5050      NORMAL :
        NEXT : PRINT :
      NEXT : PRINT "TASTE DRUECKEN ->";:
      GET A$: RETURN
6000  TEXT : HOME : FLASH :
      PRINT  TAB( 6)"SIE HABEN ES GESCHAFFT
      !"
6010  PRINT : NORMAL : PRINT "HERZLICHEN
      GLUECKWUNSCH"
```

```
6020  PRINT : PRINT : PRINT "N:
      GLEICHER IRRGARTEN NOCHMAL"
6030  PRINT "E=ENDE DES PROGRAMMS"
6040  PRINT "K=NEUER IRRGARTEN" .
6050  GET A$: IF A$ = "K" THEN  CLEAR :
      HOME : GOTO 100
6060  IF A$ = "E" THEN END
6070  IF A$ <  > "N" THEN 6000
6080  GOTO 2000
```

Abbildung 45: Programm Irrgarten.

13.3 MONDPHASEN.

Das Programm in Abbildung 46 zeichnet die Mondphase für ein gegbenes Datum.

```
10  HOME :
    FOR I = 1 TO 6: PRINT  CHR$ (7):
    NEXT
15  FOR I = 1293 TO 1305:
      POKE I,32: POKE I + 512,32:
    NEXT
20  FOR I = 1421 TO 1679 STEP 128:
      POKE I,32: POKE I + 12,32:
    NEXT
25  POKE 1551,141: POKE 1553,143:
    POKE 1555,144: POKE 1557,136:
    POKE 1559,129
30  VTAB (10): PRINT "PROGRAMM ZUR BERECHNUNG
    DER MONDPHASE": PRINT
35  PRINT "AN EINEM GEWUENSCHTEN TAG":
    PRINT
40  PRINT "IN APPLESOFT FUER DEN EURO-APPLE
    II PLUS
45  VTAB (17): PRINT  TAB( 8)"COPYRIGHT
     OKTOBER 1980
50  PRINT  TAB( 9)"BY  WOLFGANG F. HAHN
```

```
100  VTAB (23): INPUT "WOLLEN SIE ERLAEUTERUNGEN?(JA/NEIN)
     ";A$
105  IF A$ = "NEIN" GOTO 500
110  IF A$ <  > "JA" GOTO 100
115  HOME
120  PRINT "SIE KOENNEN ALS GRUNDLAGE DER
     BERECH-": PRINT
125  PRINT "NUNGEN ENTWEDER DAS DATUM DES
     LETZTEN": PRINT
130  PRINT "NEUMONDES EINGEBEN ODER EIN
     PROGRAMM-": PRINT
135  PRINT "INTERNES DATUM VERWENDEN.":
     PRINT : PRINT
140  PRINT "GEBEN SIE BEIM DATUM FOLGENDES
     EIN:": PRINT
145  PRINT "  1. TAG (1 - 31)":
     PRINT
150  PRINT "  2. MONAT (1 - 12)":
     PRINT
155  PRINT "  3. JAHR (VIERSTELLIG)":
     PRINT
160  PRINT "  4. UHRZEIT (1 - 24)":
     PRINT : PRINT
165  PRINT "Z.B. 3.MAI 1981,21 UHR :
      3,5,1981,21
170  PRINT : PRINT " (WEITER MIT BELIEBIGEM
     TASTENDRUCK)..
175  VTAB (23): HTAB (40)
180  GET A$: HOME
185  PRINT "SIE KOENNEN SICH NOCH ZUSAETZLICH
     EINEN": PRINT
190  PRINT "'REFERENZKREIS' DARSTELLEN
     LASSEN,DER": PRINT
195  PRINT "PUNKTIERT DEN VOLLMOND ZEIGT.":
     PRINT
200  VTAB (23): PRINT " (WEITER MIT BELIEBIGEM
     TASTENDRUCK)...
205  VTAB (23): HTAB (40)
210  GET A$
500  HOME
505  RESTORE : DIM WN$(6)
510  READ X0,Y0,RADIUS,SM
515  READ P1,P2
```

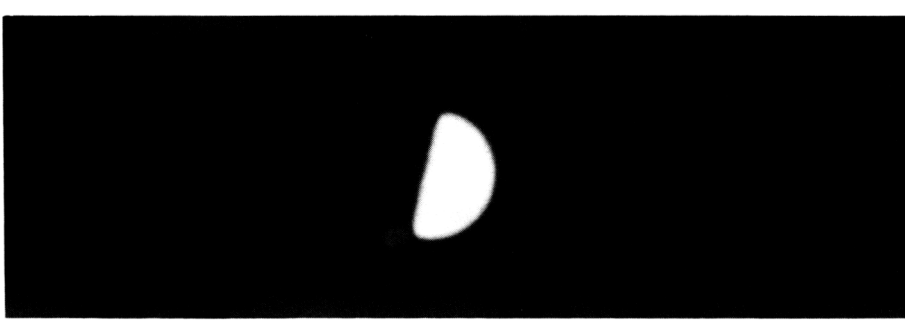
```
 520  FOR W = 0 TO 6
 525    READ WN$(W):
      NEXT
 530  DATA  140,80,75
 535  DATA  29.53
 540  DATA  6.283185307,4.712388981
 545  DATA  SONNTAG,MONTAG,DIENSTAG,MITTWOCH,
      DONNERSTAG,FREITAG,SAMSTAG
 550  RR = RADIUS ^ 2
1000  PRINT "WOLLEN SIE EIN NEUMONDDATUM
      EIN -": PRINT
1005  PRINT "GEBEN (=1) ODER DAS PROGRAMMINTERNE":
      PRINT
1010  INPUT "VERWENDEN (=2) :
       ";A
1015  PRINT : IF A = 1 GOTO 1030
1020  IF A <  > 2 GOTO 1000
1025  DN = 723111.16: GOTO 1055
1030  INPUT "DATUM NEUMOND(TAG,MONAT,JAHR,
      ZEIT): ";T,M,J,S: PRINT
1035  GOSUB 3005
1040  GOSUB 3040: IF FL = 0 GOTO 1030
1045  DN = TAGE
1050  HOME : PRINT : PRINT
1055  PRINT : PRINT "FUER WELCHES DATUM
      WOLLEN SIE DIE MOND-":
      PRINT
1060  INPUT "PHASE? (TAG,MONAT,JAHR,ZEIT)";T,
      M,J,S
1065  J1 = J: GOSUB 3005
1070  GOSUB 3040: IF FL = 0 GOTO 1055
1075  D = TAGE - DN
1500  MA = D * P1 / SM
1505  F =  COS (MA)
1510  IF  SIN (MA) < 0 GOTO 1530
1515  REM  ERSTES UND ZWEITES VIERTEL
1520  VX = 1: GOTO 2005
1525  REM  DRITTES UND LETZTES VIERTEL
1530  VX =  - 1
2000  REM  REFERENZKREIS (VOLLMONDSCHEIBE)
2005  PRINT : PRINT "WOLLEN SIE EINEN REFERENZ-":
      PRINT
2010  INPUT "KREIS (JA/NEIN)? ";A$
```

```
2015  IF A$ = "NEIN" GOTO 2200
2020  IF A$ <  > "JA" GOTO 2005
2025  HGR
2030  FOR I = 0 TO P1 STEP P2 / 18
2035    X = X0 + RADIUS *  COS (I):
        Y = Y0 + RADIUS *  SIN (I)
2040    HPLOT X,Y
2045  NEXT
2050  GOTO 2205
2200  HGR
2205  VTAB (21)
2210  FOR I =  - 75 TO 75
2215    Y =  ABS (I):YP = I + 80
2220    X = VX *  SQR (RR - Y ^ 2)
2225    XK = X0 + X:XE = X0 + F * X
2230    HPLOT XK,YP TO XE,YP
2235  NEXT
2240  VTAB (21)
2245  PRINT "MONDPHASE AM ";WN$(W);",DEN
      ";T;". ";M;". ";J1;"
2250  PRINT "UM ";S;" UHR"
2255  INPUT "WOLLEN SIE NOCHMALS (JA/NEIN)?
      ";A$
2260  IF A$ = "JA" GOTO 1055
2265  IF A$ <  > "NEIN" GOTO 2255
2270  HGR : TEXT : HOME
2275 END
3000  REM  UNTERPROGRAMM DATUM
3005  TAGE = 0
3010  TAGE = (T - 1) + 31 * (M - 1) + 365
      * (J - 1)
3015  IF M > 2 GOTO 3025
3020  J = J - 1
3025  TAGE = TAGE -  INT (.4 * M + 2.3)
      +  INT (J / 4) + S / 24
3030  W =  INT (TAGE -  INT (TAGE / 7) * 7)
3035  RETURN
3040  IF TAGE < 693699 OR TAGE > 766715
      GOTO 3050
3045  FL = 1: RETURN
3050  PRINT : PRINT : PRINT  CHR$ (7)
3055  PRINT "UNZULAESSIGES DATUM":
      PRINT
```

```
3060  FL = 0: RETURN
```

Abbildung 46: Programm Mondphase.

Abbildung 47: Beispiel Mondphase.



Originalaufname des Mondes zum gleichen Zeitpunkt.

13.4 UNORDNUNG.

Das in Abbildung 48 angegebnene Programm zeichnet das Muster in Abbildung 49.

```
  5  REM  PROGRAMM UNORDNUNG
 10  HGR2 : HCOLOR= 3
100  FOR XP = 10.5 TO 271 STEP 10
110    FOR YP = 10.5 TO 181 STEP 10
120      X = XP - 140:Y = YP - 95:
         D =  SQR (X * X + Y * Y * 2)
125      W = ( RND (1) * 2 - 1) * (170
         - D) / 500
130      IF D < 10 THEN D = 10
135      D = 200 / D: IF D < 2 THEN D = 0
136      IF D > 15 THEN D = 15
140      X0 = XP + ( RND (1) * 2 - 1) * D:
         Y0 = YP + ( RND (1) * 2 - 1) * D
200      C =  COS (W):S =  SIN (W)
300      L = 0:X =  - 5:
         Y =  - 5: GOSUB 1000:
         X = 5:Y =  - 5: GOSUB 1000
310      X = 5:Y = 5: GOSUB 1000:
         X =  - 5:Y = 5: GOSUB 1000
320      X =  - 5:Y =  - 5:
         GOSUB 1000
500    NEXT :
     NEXT :END
1000 X1 = X * C + Y * S:
     Y1 = Y * C - X * S
1020 X1 = X1 + X0:Y1 = Y1 + Y0
1100 IF L = 0 THEN  HPLOT X1,Y1:
     L = 1: RETURN
1110 HPLOT  TO X1,Y1: RETURN
```

Abbildung 48: Programm Unordnung.

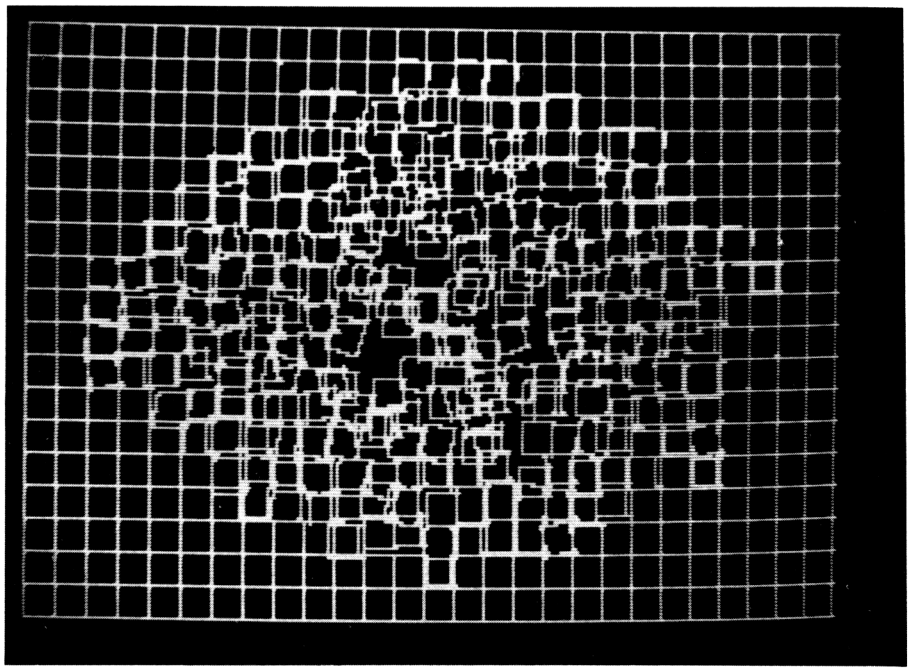


Abbildung 49: Unordnung.

Artikel-verwaltung 14

Das Programm Artikelverwaltung besteht aus 2 Teilprogrammen AVV und ZUS. Das Teilprogramm AVV dient zur Aufstellung der Eingabedatei, das Teilprogramm ZUS bindet mehrere Dateien zu einer Datei zusammen.

14.1 Programmbeschreibung AVV.

Nach dem Start des Programms mit RUN erscheint das Menü

```
*** EINGABEDATEI ***

N) EUE EINGABE
F) ORTSETZEN
L) OESCHEN
D) RUCKEN
A) BLEGEN
B) EENDEN

WAS?
```

N)eue Eingabe

Mit N wird eine neue Eingabedatei eröffnet. Ist eine alte Eingabedatei vorhanden, so wird das Datum dieser Datei eingeblendet und angefragt, ob diese Datei gelöscht werden kann. Wird dies mit J beantwortet, so wird diese Datei gelöscht und nach dem neuen Datum gefragt. Nach dieser Eingabe beginnt die Einzeleingabe der Daten.

Die Eingabezeile hat folgendes Format:

```
_____  __________  _______  ________
MENGE  BESTELL-NR. EINZEL-  GESAMT-
                   PREIS
EINGABE:_____
```

Die Eingabe erfolgt in die Eingabezeile. Die Anzahl der Striche gibt die Zahl der möglichen Stellen an. Was eingegeben werden soll, wird in der Eingabezeile invers dargestellt. Intern wird jede Eingabe als Zeichenkette gespeichert, die bei notwendigen numerischen Berechnungen in einen numerischen Wert umgewandelt wird.

Die Mengenangabe darf daher nur die Zahlen 0-9, keine Buchstaben enthalten. Die Zahl wird automatisch rechtsbündig eingesetzt. Die Bestellnummer kann eine Zahl oder eine Bezeichnung sein. Die maximale Zahl der Stellen beträgt 11. Der Einzelpreis muß mit Dezimalpunkt eingegeben werden. Also .98 oder 1.10 oder 10.00. Die Eingabe von 12 ohne Dezimalpunkt wird als 12.00 interpretiert.

Jede einzelne Eingabe wird mit der RETURN-Taste abgeschlossen. Nach der Berechnung des Gesamtpreises erscheint in der Eingabezeile die Anfrage OK?. Nur die Taste N löscht die Eingabe, alle anderen Tasten schreiben die Zeile in die Eingabedatei. Die Zeile wird auf dem Bildschirm eine Stelle hochgeschoben und bleibt sichtbar.

Das entgültige Abschließen der Eingabe erfolgt bei der Mengeneingabe durch Eingabe einer leeren Zeichenkette, also nur RETURN, ohne vorangehendes Zeichen.

Die Eingabedatei wird dabei noch nicht auf Diskette gespeichert, sondern erst, wenn mit B das Programm beendet wird. Bei längeren Eingaben empfiehlt es sich die Eingabe und das Programm zwischendurch zu beenden, um die bisherigen Eingaben auf Diskette zu speichern.

Mit F kann die Eingabe fortgesetzt werden. Ist keine Ein-

gabedatei im Rechner, so wird sie von der Diskette eingelesen. Die Eingabe wird in der üblichen Weise beendet.

Die Eingaben werden nach Bestellnummern sortiert. Ist eine Bestellnummer schon in der Datei vorhanden, so wird die neue Eingabe hinzuaddiert. Dabei wird der neue Einzelpreis übernommen.

In dieser Eingabedatei kann auch ein Eintrag gelöscht werden. Dies geschieht durch Eingabe von L und der Bestellnummer. Wird diese nicht gefunden, so erscheint eine Fehlermeldung.

Sind alle Eingaben gemacht worden, so wird die Eingabedatei mit der Eingabe von A abgelegt. Der Name dieser neuen Datei setzt sich aus dem Wort DATEI und dem Datum zusammen. Die Eingabedatei vom 1.10.83 wird als DATEI-1.10.83 auf der Diskette in Drive 1 oder 2 nach Eingabe dieser Zahl gespeichert.

Achtung:

Eine schon vorhandene Datei gleichen Namens wird dabei gelöscht.

Nochmal zurück zum Programmteil NEUE EINGABE.

Wird durch Eingabe von N dieser Programmteil angesprungen so wird, falls eine Eingabedatei noch auf der Diskette vorhanden ist, angefragt, ob diese gelöscht werden kann. Wird dies mit J beantwortet, so wird nach dem neuen Datum gefragt. Dieses Datum ist intern im Rechner ebenfalls als Zeichenkette gespeichert. Dadurch können ohne weiteres Zusätze zum Datum gemacht werden. Wenn also mehrere Eingabedateien unter dem gleichen Datum angelegt werden, so kann dieses Datum durch Zusätze erweitert werden, z.B. DATUM:2.10.81/A oder DATUM:2.10.81/Müller&CO.

Beim Ablegen der Datei aif eine andere Diskette können so Dateien mit gleichen Datum unterschieden werden.

Eine Druckausgabe wird mit D eingeleitet. Sie kann auf den Bildschirm oder einen Drucker augegeben werden. Die

Interfacekarte für den Drucker muß in SLOT 1 des Appel-Bus stecken. Die Druckausgabe ist nicht formatiert. Es wird nur die gesamte Zeichenkette ausgegeben.

Eine Formatierung muß dem jeweiligen.Format angepasst werden. Die einzelnen Elemente sind auf folgende Weise aus der Zeichenkette erhältlich (i-ter Eintrag).

Die Menge mit	MID$(I$(I),12,6)
die Bestellmenge mit	LEFT$(I$(I),11)
der Einzelpreis mit	MID$(I$(I),19,8)
der Gesamtpreis mit	RIGHT$(I$(I),9)

Damit kann in Zeile 6120 eine eigene Druckausgabe programmiert werden.

Treten Fehler auf, die nicht durch das Programm abgefangen werden, so kann in den meisten Fällen mit einem GOTO 2000 das Programm neu gestartet werden, ohne daß bisherige Eingaben verloren gehen.

Die Eingabedatei hat im Programm den Namen I$. Sie hat maximal die Länge von 200 Einträgen. Die Zahl der Einträge (N) wird als erste Zahl in die gespeicherte Datei geschrieben. Dadurch kann sie auch in anderen Programmen, zum Beispiel Ausschreiben von Lieferscheinen verwendet werden.

Folgendes Programm liest eine vom Programm AVV oder ZUS angelegte Textdatei:

```
9000 D$ =  CHR$ (4)
9010  INPUT "DATEINAME:";F$
9020  INPUT "DRIVE:";I
9030  PRINT D$;"OPEN";F$;",D";I
9040  PRINT D$;"READ";F$
9050  INPUT N
9060  FOR J = 0 TO N - 1
9070  INPUT I$(J)
9080  NEXT J
9090  PRINT D$;"CLOSE";F$
9999  END
```

Das folgende Programm addiert alle Gesamtpreise zu einer Gesamtsumme zusammen. Es benötigt vom Hauptprogramm das Unterprogramm 4800 - 4870.

```
9100 GP = 0
9110  FOR J = 0 TO N - 1
9115 P =  VAL ( RIGHT$ (I$(J),9))
9120 GP = GP + P
9130  NEXT J
9140  GOSUB 4800
9999  END
```

14.2 Programmteil ZUS.

Die in dem Programm AVV abgelegten Dateien können mit dem Programm ZUS zu neuen Dateien zusammengesetzt werden. Nach RUN ZUS erscheint das Menü:

```
*** ZUSAMMENFUEGEN ***

          N)EUE DATEI
          E)INFUEGEN
          K)ATALOG
          A)USDRUCKEN
          L)OESCHEN
          B)EENDEN

          WAS ?
```

Mit N wird eine neue Datei angelegt. Nach Eingabe des Dateinamens und der Drive-Nummer prüft das Programm, ob eine entsprechende Datei schon vorhanden ist. Dann erscheint eine entsprechende Fehlermeldung. Ist der Name nicht vorhanden, so wird eine leere Datei, mit N=0 angelegt.

In diese Datei kann mit E der Inhalt einer anderen Datei eingefügt werden. Dazu wird entweder der N)ame oder das

D)atum der Datei angegeben welche eingefügt werden soll und danach der N)ame oder das D)atum der Zieldatei.

Beim Zusammenbau der Daten werden bei gleichen Einträgen die Mengen und die Gesamtpreise addiert. Mit K wird ein Katalog der Disketten in Drive 1 oder 2 ausgegeben

Für den Programmteil A)USDRUCKEN gilt genau das gleiche wie für das Drucken im Programm AVV. Die Druckausgabe ist hier in Zeile 3100 programmiert. Die Datei hat hier die Bezeichnung ART$.

Mit L)OESCHEN wird eine Datei auf der Diskette gelöscht. Diese Datei wird auf beiden Laufwerken (wenn vorhanden) gesucht und zur Bestätigung nochmals angezeigt. Mit J)A wird sie gelöscht.

Das Programm wird mit B beendet.

Allgemeine Bemerkungen zum Programm ZUS:

Ist nur ein Laufwerk angeschlossen, so sollte Zeile 5250 in 5250 POKE 216,0 geändert werden. Wird mit 2 Laufwerken gearbeitet, wobei Programm und Eingabedatei über Laufwerk 1 und abgelegte Dateien auf Laufwerk 2 gehalten werden, so sollte Zeile 5205 in 5205 I=2 und Zeile 5250 in IF I=2 THEN I=1:POKE 216,0: GOTO 5210 geändert werden.

Wird statt einer verlangten Eingabe eine leere Eingabe gemacht, also nur die RETURN-Taste betätigt, so wird in das Menü zurückgesprungen. Ein abgestütztes Programm kann mit GOTO 2000 neu gestartet werden.

Von den Datendisketten sollten unbedingt Backup`s gemacht werden, da bei einem Systemzusammenbruch alle eingegebenen Daten verloren gehen.

```
100  CALL  - 936: GOSUB 600:
     GOTO 2000
600  REM  ANFANGSWERTE
605  N = 0:X1 = 0:S2 =  - 1:
     X2 =  - 1
610  POKE 33,40
620  DIM EN(8),E$(7),B$(4),I$(200)
630  EN(1) = 6:EN(2) = 11:
     EN(3) = 8:EN(4) = 9
640  EN(5) = 1:EN(6) = 9:
     EN(7) = 22:EN(8) = 31
650  E$(1) = "MENGE":E$(2) = "BESTELL-NR."
655  E$(3) = "EINZEL-":
     E$(4) = "GESAMT-"
660  E$(5) = "PREIS"
665  E$(6) = "             ":
     E$(7) = "DATEI ":F$ = ""
670  D$ =  CHR$ (4):DA$ = ""
675  BEEP$ = ""
680  PRINT D$;"NOMON C,I,O"
699  RETURN
700  REM  DATUM
710  ONERR  GOTO 800
720  PRINT D$;"VERIFY DATE,D1"
730  POKE 216,0
740  PRINT D$;"OPEN DATE,D1"
750  PRINT D$;"READ DATE"
760  INPUT DA$
770  PRINT D$;"CLOSE DATE"
780  VTAB 3: HTAB 9: PRINT "DATUM:
     ";DA$
790  RETURN
800  VTAB 4: HTAB 3: INPUT "NEUES DATUM:
     ";DA$: PRINT D$
810  RRINT D$;"OPEN DATE,D1"
820  PRINT D$;"WRITE DATE"
830  PRINT DA$
840  PRINT D$;"CLOSE DATE "
850  VTAB 4: CALL  - 868
860  VTAB 3: HTAB 9: PRINT "DATUM:
     ";DA$
2000 REM  EINGABE
2005 CALL  - 936
```

```
2007  VTAB 1: HTAB 9: PRINT "*** EINGABEDATEI
      ***"
2008  IF DA$ <  > "" THEN  VTAB 3:
      HTAB 9: PRINT "DATUM:
      ";DA$
2010  VTAB 4
2020  HTAB 27: PRINT "N)EUE EINGABE"
2030  HTAB 27: PRINT "F)ORTSETZEN"
2035  HTAB 27: PRINT "L)OESCHEN"
2037  HTAB 27: PRINT "D)RUCKEN"
2038  HTAB 27: PRINT "A)BLEGEN"
2040  HTAB 27: PRINT "B)EENDEN"
2050  VTAB 11: HTAB 27: PRINT "WAS?";:
      GET A$: PRINT D$
2060  IF A$ = "N" THEN  GOTO 3000
2065  IF A$ = "D" THEN  GOSUB 6000:
      GOTO 2000
2070  IF A$ = "F" THEN  GOTO 3200
2080  IF A$ = "B" THEN  GOSUB 3500:
     END
2090  IF A$ = "L" THEN  GOTO 2100
2095  IF A$ = "A" THEN  GOSUB 7000:
      GOTO 2000
2099  PRINT BEEP$: GOTO 2000
2100  REM  EINTRAG LOESCHEN
2105  IF N = 0 THEN  GOSUB 3600:
      GOSUB 700
2110  VTAB 16: HTAB 2: PRINT "WELCHE BESTELL-NR.?
2120  VTAB 17: HTAB 21: PRINT "----------";:
      HTAB 21: VTAB 16: INPUT "";A$
2130  B$(2) =  LEFT$ (A$ + E$(6),11):
      GOSUB 2500: GOTO 2000
2500  REM  SUCHEN
2510  J = 0:C$ =  LEFT$ (I$(0),11)
2520  IF B$(2) < C$ THEN 2599
2525  J1 = 0:J2 = N
2530  J =  INT ((J1 + J2) / 2):
      C$ =  LEFT$ (I$(J),11)
2535  IF B$(2) = C$ THEN  GOTO 2600
2540  IF B$(2) < C$ THEN J2 = J:
      GOTO 2550
2545  J1 = J
2550  IF J <  >  INT ((J1 + J2) / 2) THEN 2530
```

```
2599  VTAB 20: HTAB 2: PRINT "KEINE BESTELL-NR.";B$(2);"
      GEFUNDEN.": GOSUB 6900:
      GOTO 2000
2600  REM  EINTRAG GEFUNDEN
2610  IF J > N THEN 2690
2615  I$(J) = I$(J + 1):
      J = J + 1: GOTO 2610
2690  N = N - 1:X2 = 1: RETURN
3000  REM  EINGABEDATEI VORHANDEN ?
3010  GOSUB 700: POKE 216,0
3015  ONERR  GOTO 4000
3020  PRINT D$;"VERIFY EINGABEDATEI,D1"
3025  VTAB 18: HTAB 1: PRINT "SOLL VORHANDENE
      EINGABEDATEI";
3035  VTAB 19: HTAB 1: PRINT "GELOESCHT
      WERDEN? (J/N) ";: GET A$:
      PRINT D$;:
3040  IF A$ = "N" THEN 2000
3050  PRINT D$;"DELETE EINGABEDATEI,D1"
3055  N = 0
3060  POKE 216,0
3070  VTAB 12: HTAB 1: CALL  - 958
3080  GOSUB 800: GOTO 4000
3200  REM  EINGABE FORTSETZEN
3210  IF N = 0 THEN  GOSUB 3600:
      GOSUB 700
3230  GOTO 4000
3500  REM  EINGABEDATEI SCHREIBEN
3505  IF X2 < 1 THEN 3550
3510  PRINT D$;"OPEN EINGABEDATEI,D1"
3515  PRINT D$;"WRITE EINGABEDATEI"
3520  PRINT N
3530  FOR I = 0 TO N - 1:
        PRINT I$(I):
      NEXT I
3540  PRINT D$;"CLOSE EINGABEDATEI"
3550  RETURN
3600  REM  EINGABEDATEI LESEN
3604  ONERR  GOTO 9992
3610  PRINT D$;"OPEN EINGABEDATEI,D1"
3615  PRINT D$;"READ EINGABEDATEI"
3620  INPUT N
```

```
3630  FOR I = 0 TO N - 1:
        INPUT I$(I):
      NEXT I
3640  PRINT D$;"CLOSE EINGABEDATEI"
3650  X1 = 1: RETURN
4000  REM  EINGABE
4005  H = 18:X2 = 1
4010  VTAB 24: HTAB 5: PRINT "BEENDEN DER
      EINGABE MIT <RET>";
4025  VTAB H - 1: HTAB 1:
      PRINT "------  ----------  --------
      ---------"
4030  VTAB H: HTAB 9: PRINT E$(2);:
      HTAB 22: PRINT E$(3);:
      HTAB 31: PRINT E$(4)
4040  PRINT : HTAB 28: PRINT E$(5)
4100  I = 1: VTAB H: HTAB 1:
      INVERSE : PRINT E$(1):
      NORMAL
4110  VTAB H + 4: HTAB 2:
      PRINT "EINGABE:";: CALL  - 868
4112  VTAB H + 5: CALL  - 868:
      FOR J = 1 TO EN(I):
        PRINT "-";:
      NEXT J
4114  VTAB H + 4: HTAB 10:
      INPUT "";A$: PRINT D$
4115  IF A$ = "" THEN 2000
4120  L =  LEN (A$): IF L > EN(I) THEN 4110
4125  B$(I) = A$
4127  IF I = 2 THEN  VTAB H - 2:
      HTAB EN(6): PRINT A$:
      GOTO 4140
4130  L = EN(I) - L + EN(I + 4):
      VTAB H - 2: HTAB L:
      PRINT A$
4140  VTAB H: HTAB EN(I + 4):
      PRINT E$(I);
4150  I = I + 1: IF I = 4 THEN 4500
4160  HTAB EN(I + 4): INVERSE :
      PRINT E$(I): NORMAL :
      GOTO 4110
4500  GP =  VAL (B$(1)) *  VAL (B$(3))
```

```
4505  GOSUB 4800:B$(4) = GP$
4510  L =  LEN (GP$):L = EN(4) - L + EN(8):
      HTAB L: VTAB H - 2:
      PRINT GP$
4520  VTAB H + 4: HTAB 10:
      CALL  - 868: PRINT "OK?";:
      GET A$: PRINT D$
4530  IF A$ = "N" THEN  VTAB H - 2:
      HTAB 1: CALL  - 868:
      GOTO 4100
4540  POKE 34,12: POKE 35,16:
      CALL  - 912
4545  POKE 35,24: POKE 34,0
4550  GOSUB 4700: IF N = 0 THEN I$(0) = V$:
      GOTO 4605
4600  GOSUB 5000: GOSUB 5200
4605  N = N + 1: IF N > 499 THEN  GOTO 9991
4610  GOTO 4100
4700  V$ =  LEFT$ (B$(2) + E$(6),11)
4705  B$(2) = V$
4710  I = 1:K = 6: GOSUB 4880
4720  I = 3:K = 8: GOSUB 4880
4730  I = 4:K = 9: GOSUB 4880
4740  RETURN
4800  GP$ =  STR$ (GP):L =  LEN (GP$)
4810  FOR J = 1 TO L
4820    IF  MID$ (GP$,J,1) = "." THEN  GOTO 4840
4830  NEXT J
4840  IF L - J = 2 THEN 4870
4850  IF L - J = 1 THEN GP$ = GP$ + "0":
      GOTO 4870
4860  IF L - J =  - 1 THEN GP$ = GP$ + ".00"
4870  B$(4) = GP$: RETURN
4880  C$ = B$(I):L =  LEN (C$):
      L = K - L:C$ =  LEFT$ (E$(6),L) + C$:
      V$ = V$ + C$: RETURN
5000  J = 0
5010  C$ =  LEFT$ (I$(0),11)
5020  IF B$(2) < C$ THEN 5100
5030  J1 = 0:J2 = N
5040  J =  INT ((J1 + J2) / 2)
5045  C$ =  LEFT$ (I$(J),11)
5050  IF B$(2) = C$ THEN 5500
```

```
5060  IF B$(2) < C$ THEN J2 = J:
      GOTO 5080
5070  J1 = J
5080  IF J <  > INT ((J1 + J2) / 2) THEN 5040
5090  J = J + 1
5100  RETURN
5200  J1 = N
5210  IF J1 < J + 1 THEN 5290
5220  I$(J1) = I$(J1 - 1)
5230  J1 = J1 - 1: GOTO 5210
5290  I$(J) = V$
5299  RETURN
5500  REM  DOPPELTER EINTRAG
5510  J1 = J:C$ =  MID$ (I$(J1),12,6)
5520  M =  VAL (B$(1)) +  VAL (C$):
      B$(1) =  STR$ (M)
5530  GP =  VAL (B$(3)) * M:
      GOSUB 4800:B$(4) = GP$:
      GOSUB 4700
5540  J = J1:I$(J) = V$
5550  POP : GOTO 4100
6000  REM  AUSDRUCKEN
6010  CALL  - 936
6020  IF N = 0 THEN  GOSUB 3600:
      GOSUB 700
6030  VTAB 10:
6035  HTAB 27: PRINT "B)ILDSCHIRM"
6040  HTAB 27: PRINT "D)RUCKER"
6050  VTAB 13: HTAB 27: PRINT "WAS?";:
      GET A$: PRINT D$
6060  IF A$ = "D" THEN  PRINT D$;"PR#1":
      B1% = 1
6070  CALL  - 936
6080  VTAB 2: HTAB 9: PRINT "DATUM:
      ";DA$
6090  PRINT
6100  J = 0
6110  FOR I = 0 TO N - 1
6120    PRINT I$(I)
6130    IF B1% = 1 THEN 6150
6140    J = J + 1: IF J = 18 THEN  GOSUB 6900:
        J = 0
6150  NEXT I
```

```
6160  IF Bl% = 1 THEN Bl% = 0:
      PRINT D$;"PR#0"
6170  GOSUB 6900
6180  RETURN
6900  VTAB 24: HTAB 1: INVERSE :
      PRINT "WEITER MIT <SPACE>-TASTE";
6905  NORMAL : GET A$: PRINT D$
6910  RETURN
7000  REM  ABLEGEN
7005  IF N = 0 THEN  GOSUB 3600:
      GOSUB 700
7010  VTAB 12: HTAB 1: CALL  - 958
7020  VTAB 16: PRINT "EINGABEDATEI WIRD
      ALS"
7030  VTAB 18: HTAB 9: INVERSE :
      PRINT "DATEI";DA$: NORMAL
7040  VTAB 20: PRINT "AUF DRIVE 1) ODER
      2)  ABGELEGT";: HTAB 21:
      GET I: PRINT D$
7050  ONERR  GOTO 7900
7060  PRINT D$;"OPEN DATEI";DA$;",D";I
7070  PRINT D$;"WRITE DATEI";DA$
7075  PRINT N
7080  FOR J = 0 TO N - 1
7090    PRINT I$(J)
7100  NEXT J
7110  PRINT D$;"CLOSE DATEI";DA$
7120  POKE 216,0
7130  RETURN
7900  Y =  PEEK (222)
7910  IF Y = 8 THEN  VTAB 22:
      PRINT "KEINE DISKETTE IN DRIVE 2:
      GOSUB6900:GOTO 2000
7920  VTAB 22: PRINT "FEHLER":
      GOSUB 6900: GOTO 2000
```

Abbildung 50: Programm AVV.

```
  10  REM  ************************
  12  REM  * PROGRAMMTEIL ZUS     *
  14  REM  * E.FLOEGEL 12.11.81   *
  16  REM  * VERSION 1.31         *
  18  REM  ************************
 600  REM  **********************
 601  REM  *** ANFANGSWERTE  ***
 602  REM  **********************
 610  BEEP$ = ""
 615  DIM ART$(200): DIM H$(200):
      DIM H1$(400)
 620  D$ =  CHR$ (4):BL$ = "           ":
      REM  11 BLANKS
 630  N = 0:B4% = 0
 640  H = 18:V = 4
 650  PRINT D$;"NOMON C,I,O"
 660  MAX = 200
 670  POKE 33,40
 680  S1% = 0:S2% = 0:S3% = 0
 690  CALL  - 936: GOTO 2000
2000  REM  **********************
2010  REM  *** VERTEILER    ***
2020  REM  **********************
2030  CALL  - 936: VTAB 1:
      HTAB 9: PRINT "*** ZUSAMMENFUEGEN ***"
2040  VTAB 5
2060  HTAB 27: PRINT "N)EUE DATEI"
2070  HTAB 27: PRINT "E)INFUEGEN"
2080  HTAB 27: PRINT "K)ATALOG"
2090  HTAB 27: PRINT "A)USDRUCKEN"
2100  HTAB 27: PRINT "L)OESCHEN"
2105  HTAB 27: PRINT "B)EENDEN"
2110  VTAB 13: HTAB 27: PRINT "WAS?";:
      GET A$: PRINT D$
2120  IF A$ = "N" THEN  GOSUB 4000:
      GOTO 2030
2130  IF A$ = "B" THEN 9999
2140  IF A$ = "K" THEN  GOSUB 5500:
      GOTO 2030
2150  IF A$ = "L" THEN  GOTO 5600
2160  IF A$ = "E" THEN  GOTO 4100
2170  IF A$ = "A" THEN  GOTO 4900
2180  PRINT BEEP$: GOTO 2110
```

```
3000  REM  *********************
3010  REM  *** AUSDRUCKEN    ***
3020  REM  *********************
3025  GOSUB 3030: GOTO 3070
3030  CALL  - 936
3040  VTAB 10:
3045  HTAB 27: PRINT "B)ILDSCHIRM"
3050  HTAB 27: PRINT "D)RUCKER"
3055  VTAB 13: HTAB 27: PRINT "WAS?";:
      GET A$: PRINT D$
3060  IF A$ = "D" THEN  PRINT D$;"PR#1":
      B3% = 1: GOTO 3068
3062  IF A$ = "" THEN  PRINT BEEP$:
      GOTO 3055
3065  CALL  - 936
3067  IF A$ <  > "B" THEN  PRINT BEEP$:
      GOTO 3055
3068  RETURN
3070  VTAB 2: HTAB 9: PRINT F1$
3075  PRINT
3080  J = 0
3090  FOR I = 0 TO N - 1
3100    PRINT ART$(I)
3105    IF B3% = 1 THEN 3115
3110    J = J + 1: IF J = 18 THEN  GOSUB 3900:
        J = 0
3115  NEXT I
3125  IF B3% = 1 THEN B3% = 0
3126  PRINT D$;"PR#0"
3127  GOSUB 3900
3130  RETURN
3900  VTAB 24: HTAB 9: INVERSE :
      PRINT "WEITER MIT <SPACE>-TASTE";:
      NORMAL : GET A$: PRINT D$;:
      GOSUB 3910: RETURN
3910  HTAB 22: VTAB 1: CALL  - 868:
      HTAB 24: VTAB 1: CALL  - 868:
      RETURN
3995  VTAB 22: HTAB 1: INVERSE :
      PRINT "FEHLER";: NORMAL :
      PRINT "     ";: RETURN
4000  REM  *********************
4001  REM  ***   NEUE DATEI  ***
```

```
4002  REM  **********************
4010  VTAB 15: HTAB 2: INPUT "NEUER DATEINAME:
      ";F$: PRINT D$
4015  IF A$ = "" THEN 2030
4020  HTAB 2: PRINT "AUF DRIVE 1 ODER 2
      ?";: GET I: PRINT D$
4030  GOSUB 5300
4040  N = 0: GOSUB 5400:
      CALL  - 936: GOTO 2030
4100  REM  **********************
4101  REM  ***  EINFUEGEN    ***
4102  REM  **********************
4103  : GOSUB 4105: GOTO 4130
4105  VTAB 15: HTAB 2: PRINT "WELCHE DATEI?
      D)ATUM ODER N)AME";:
      GET A$: PRINT D$
4107  IF A$ = "" THEN 2030
4110  VTAB 15: HTAB 2: CALL  - 868
4115  IF A$ = "D" THEN  INPUT "DATUM:
      ";A$: PRINT D$:F1$ = "DATEI" + A$:
      GOTO 4122
4120  IF A$ = "N" THEN  INPUT "NAME:
      ";F1$: PRINT D$
4122  IF (A$ = "") OR (F1$ = "") THEN 2030
4125  RETURN
4130  F$ = F1$:S1% = 1: GOSUB 5200
4140  I1 = I:B1% = 0:S1% = 0:
      GOSUB 5700
4200  VTAB 17: HTAB 2: PRINT "IN WELCHE
      DATEI? D)ATUM ODER N)AME";:
      GET A$: PRINT D$
4210  VTAB 17: HTAB 2: CALL  - 868
4215  GOSUB 4115
4220  F$ = F1$:F2$ = F1$:
      S1% = 2: GOSUB 5200
4230  I2 = I:B2% = 0:S1% = 0:
      GOSUB 5800
4235  PRINT BEEP$
4240  GOSUB 4500
4250  IF (B1% = 0) AND (B2% = 0) THEN  GOSUB 5900
4260  GOTO 2000
4500  REM  *** DATEI MERGE ***
4502  L = 0:L1 = 0:L2 = 0
```

```
4505  POKE 216,0
4510  IF N1 = 0 THEN  FOR I = 0 TO N2 - 1:
        H1$(I) = H$(I):
      NEXT I:L = N2: RETURN
4520  IF N2 = 0 THEN  FOR I = 0 TO N1 - 1:
        H1$(I) = ART$(I):
      NEXT I:L = N1: RETURN
4530  A$ =  LEFT$ (ART$(L1),11):
      B$ =  LEFT$ (H$(L2),11)
4540  IF A$ < B$ THEN H1$(L) = ART$(L1):
      L = L + 1:L1 = L1 + 1:
      GOTO 4570
4550  IF A$ = B$ THEN  GOSUB 4800:
      GOTO 4570
4560  H1$(L) = H$(L2):L = L + 1:
      L2 = L2 + 1
4570  IF (L1 < N1) AND (L2 < N2) THEN 4530
4580  IF L1 = N1 THEN  GOTO 4590
4582  H1$(L) = ART$(L1):
      L = L + 1:L1 = L1 + 1
4584  IF L1 = N1 THEN  RETURN
4586  GOTO 4582
4590  IF L2 = N2 THEN  GOTO 4599
4592  H1$(L) = H$(L2):L = L + 1:
      L2 = L2 + 1:
4594  IF L2 = N2 THEN  RETURN
4596  GOTO 4592
4599  RETURN
4600  GP$ =  STR$ (GP):L =  LEN (GP$)
4610  FOR J = 1 TO L
4620    IF  MID$ (GP$,J,1) = "." THEN 4640
4630  NEXT J
4640  IF L - J = 2 THEN 4670
4650  IF L - J = 1 THEN GP$ = GP$ + "0":
      GOTO 4670
4660  IF L - J =  - 1 THEN GP$ = GP$ + ".00"
4670  B$ = GP$: RETURN
4800  REM  *** GLEICHER EINTRAG ***
4810  A =  VAL ( MID$ (ART$(L1),12,6)):
      B =  VAL ( MID$ (H$(L2),12,6))
4820  B$ =  STR$ (A + B):
      L% =  LEN (B$): IF L% = 6 THEN 4830
```

```
4825  B$ =  LEFT$ (BL$,6 - L%) + B$
4827  A$ = A$ + B$
4830  B$ =  MID$ (ART$(L1),18,8)
4835  A$ = A$ + B$
4840  A =  VAL ( RIGHT$ (ART$(L1),9)):
      B =  VAL ( RIGHT$ (H$(L2),9))
4845  GP = A + B: GOSUB 4600
4850  L% =  LEN (B$): IF L% = 9 THEN 4898
4860  B$ =  LEFT$ (BL$,9 - L%) + B$
4898  H1$(L) = A$ + B$:L = L + 1:
      L1 = L1 + 1:L2 = L2 + 1
4899  RETURN
4900  REM  *** AUSDRUCKEN ***
4905  GOSUB 4105:F$ = F1$:
      S1% = 3: GOSUB 5200:
      I1 = I
4906  S1% = 0:I1 = I: GOSUB 5700
4907  :N = N1
4910  GOSUB 3000: GOTO 2000
5000  REM  ********************
5001  REM  ***   DISK I/O   ***
5002  REM  ********************
5010  IF B4% = 0 THEN  RETURN
5020  PRINT D$;"OPEN DATEI";DA$;",D2"
5022  PRINT D$;"DELETE DATEI";DA$;",D2"
5024  PRINT D$;"OPEN DATEI";DA$;",D2"
5030  PRINT D$;"WRITE DATEI";DA$
5040  PRINT N
5050  FOR I = 0 TO N - 1
5060    PRINT ART$(I):
      NEXT I
5070  PRINT D$;"CLOSE"
5080  RETURN
5100  REM  ***  LESEN ***
5110  PRINT D$;"OPEN DATEI";DA$;",D2"
5120  PRINT D$;"READ DATEI";DA$
5130  INPUT N
5140  FOR I = 0 TO N - 1
5150    INPUT ART$(I):
      NEXT I
5160  PRINT D$;"CLOSE"
5170  RETURN
5200  REM  *** DATEI VORHANDEN ***
```

```
5205  I = 1
5210  ONERR  GOTO 5250
5220  PRINT D$;"VERIFY";F$;",D";I
5245  RETURN
5250  IF I = 1 THEN I = 2:
      POKE 216,0: GOTO 5210
5252  Y =  PEEK (222)
5255  IF Y = 6 THEN  GOSUB 3995:
      PRINT F$;" NICHT VORHANDEN":
      GOSUB 3900: POKE 216,0:
      GOTO 2030
5260  IF Y = 8 THEN  GOSUB 3995:
      PRINT "KEINE DISKETTE IN DRIVE  ":
      GOSUB 3900
5270  POKE 216,0
5275  S% = S1%: IF S% = 0 THEN 5285
5280  ON S% GOTO 4140,4230,4906,5614
5285  GOTO 2000
5300  REM  *** NEUE DATEI ***
5310  ONERR  GOTO 5350
5320  PRINT D$;"VERIFY";F$;",D";I
5330  VTAB 22: HTAB 9: PRINT F$;" SCHON
      VORHANDEN": GOSUB 3900:
      GOTO 2030
5350  POKE 216,0: GOTO 4040
5400  ONERR  GOTO 5252
5405  PRINT D$;"OPEN";F$;",D";I
5410  PRINT D$;"WRITE";F$
5420  PRINT N
5430  PRINT D$;"CLOSE"
5440  RETURN
5500  REM  *** KATALOG ***
5510  VTAB 15: HTAB 2: PRINT " KATALOG VON
      DRIVE 1 ODER 2 ? ";:
      GET I: PRINT D$
5512  GOSUB 3030
5515  CALL  - 936
5520  PRINT D$;"CATALOG,D";I
5525  PRINT D$;"PR#0"
5530  GOSUB 3900: RETURN
5600  REM  *** LOESCHEN ***
5610  GOSUB 4105:F$ = F1$
5612  S1% = 4: GOSUB 5200
```

```
5614  S1% = 0
5630  HTAB 2: PRINT "LOESCHEN VON ";F$;",
      D";I;" J)A";: GET A$:
      PRINT D$
5640  IF A$ <  > "J" THEN 2000
5650  PRINT D$;"DELETE";F$;",D";I
5660  GOTO 2000
5700  PRINT D$;"OPEN";F1$;",D";I1
5710  PRINT D$;"READ";F1$
5712  INPUT N1
5714  IF N1 = 0 THEN 5730
5715  IF N1 > MAX THEN NZ = MAX:
      B1% = 1: GOTO 5720
5718  NZ = N1
5720  FOR I = 0 TO NZ - 1
5725    INPUT ART$(I):
      NEXT I
5730  IF B1% = 0 THEN  PRINT D$;"CLOSE";F1$
5740  RETURN
5800  PRINT D$;"OPEN";F2$;",D";I2
5810  PRINT D$;"READ";F2$
5812  INPUT N2
5814  IF N2 = 0 THEN 5830
5815  IF N2 > MAX THEN NZ = MAX:
      B2% = 1: GOTO 5820
5818  NZ = N2
5820  FOR I = 0 TO NZ - 1
5825    INPUT H$(I):
      NEXT I
5830  IF B2% = 0 THEN  PRINT D$;"CLOSE";F2$
5840  RETURN
5900  PRINT D$;"OPEN";F2$;",D";I2
5910  PRINT D$;"WRITE";F2$
5920  PRINT L
5930  FOR I = 0 TO L - 1
5940    PRINT H1$(I):
      NEXT I
5950  PRINT D$;"CLOSE"
5960  RETURN
9999  PRINT D$;"CATALOG,D1":
     END
```

Abbildung 51: Programm ZUS.

ADREKART 15

ADREKART ist ein disketten-orientiertes Adressen-Verwaltungs-Programm. Die Adressen werden unter einem Dateinamen als Liste auf der Diskette aufgezeichnet. Parallel zu dieser Datei wird eine Indexdatei eröffnet. In dieser sind die Adressen unter ihren Schlüsselworten alphabetisch geordnet. Als Schlüsselwort wird der Familienname und der Vorname verwendet.

Außerdem wird in der Indexliste die relative Lage des Eintrags in der Hauptdatei mit aufgezeichnet. Dadurch ist ein schnelles Auffinden eines Eintrags gegeben. Unter einem Dateinamen können maximal 300 Einträge verwaltet werden.

Aufbau der Adressen:

```
Familienname: Meier          (max. 20 Zeichen)
Vorname     : Hans           (max. 10 Zeichen)
Kundennr.   : 13300          (max. 6 Zeichen)
Code        : M              (max. 5 Zeichen)
Titel       : Ing.(grad)     (max. 15 Zeichen)
Firma       : Electronic     (max. 25 Zeichen)
Strasse     : Waldstr. 16    (max. 25 Zeichen)
Stadt       : 7500 Karlsruhe(max. 25 Zeichen)
Telefon     : 0721/81 75 62  (max. 15 Zeichen)
```

Die Adresse wird folgendermaßen ausgedruckt:

```
            Herrn Ing.(grad)
            Hans Meier
            Electronic
            Waldstr. 16
      7500  Karlsruhe
```

Codeaufbau:

Die Angabe eines Codewortes dient zur Auswahl bestimmter Adressen aus der Liste. Es besteht aus maximal 5 Stellen, wobei die erste Stelle vorgegeben ist und zur Format-Steuerung verwendet wird.

Für diese Stelle sind folgende Vereinbarungen getroffen:

1. Stelle = M Druckt in der 1. Zeile Herrn
1. Stelle = W Druckt in der 1. Zeile Frau
1. Stelle = F Druckt in der 1. Zeile Firma
1. Stelle = O 1. Zeile wird nicht gedruckt.

Für die weiteren Stellen kann zur Kennzeichnung jeder Buchstabe außer "X" verwendet werden. Das Zeichen X bedeutet beliebig, d.h. diese Stelle wird beim Suchen nicht berücksichtigt.

Beispiel:

Die Adressenkartei einer Schule soll folgende Gruppen unterscheiden:
1. Lehrer, Schüler, Referendare
2. Unter-, Mittel-, Oberstufe

Der Code sieht dann folgendermaßen aus:

1. Stelle	M	(festgelegt)
	W	(festgelegt)
2. Stelle	L	Lehrer
	S	Schüler
	R	Referendare
3. Stelle	U	Unter-
	M	Mittel-
	O	Oberstufe

Bei Eingabe des Codewortes MLX werden z.B. die Adressen aller männlichen Lehrer, bei Eingabe von XSU die Adressen aller Schüler der Unterstufe ausgedruckt.

Wichtig: Je besser der Aufbau des Codewortes vorher geplant wurde, desto einfacher lassen sich später ver-

schiedene Adressengruppen ausgeben.

Man kann auch die ersten beiden Stellen der Postleitzahl in den Code übernehmen. Damit können Adressen nach Postleitgebieten ausgewählt werden.

Verwaltung von 2 verschiedenen Adressen unter einem Namen:

Sollen unter einem Namen 2 oder mehrere Adressen gehalten werden, so ist dies auf folgende Weise möglich:

Am Ende des Vornamens werden 2 Leerzeichen und danach eine Zahl eingegeben. Das Programm unterscheidet nun in der Indexliste zwischen "HANS" und "HANS 1", in der Druckausgabe wird aber nur ein Leerzeichen beachtet, der Rest wird durch den Familiennamen überschrieben.

Das bedeutet aber auch, daß das 2. Leerzeichen als Begrenzer wirkt, und somit alle Zeichen, die danach im Vornamen folgen, nicht gedruckt werden.

Programmablauf:

Das Programm ist selbsterklärend. Falsche Eingaben werden durch entsprechende Fehlermeldungen angezeigt.

Die "MENUE"-Tabelle hat folgendes Aussehen:

Angabe der Datei- und Eintragsänderungen:

Dateiveränderungen:

E Eröffnen einer neuen Datei
R Lesen einer Datei
L Liste aller Einträge
T Teil-Listen
D Drucken von Einträgen

Eintragsänderungen:

I Einfügen eines neuen Eintrags
A Ausgabe....

O Löschen eines Eintrags
C Ändern....
Q Beenden des Programms

Mit (E) wird eine neue Datei eröffnet, mit (R) eine vorhandene Datei eingelesen. Mit (L) wird eine Liste aller Einträge ausgegeben. Eine Liste umfaßt den Familiennamen, Vornamen, Code und Kundennummer.

Teillisten werden mit (T) ausgegeben, wobei entweder nach Code oder nach Kundennummern selektiert werden kann. Mit (D) werden die Adressen ausgedruckt, wobei wiederum Teile der Datei durch Wahl eines bestimmten Codes oder Gruppen von Kundennummern ausgewählt werden können.

Bei den Eintragsänderungen wir mit (I) ein neuer Eintrag eingefügt, der mit (A) zur Kontrolle ausgegeben wird. Ganze Einträge werden mit (O) gelöscht, mit (C) Teile eines Eintrags geändert.

Bei einer Änderung wird zuerst der vollständige Eintrag ausgegeben, dann vom Programm durch Ausgabe von Kundennummer:, Code:, usw. abgefragt, welche Zeile geändert werden soll. Durch Eingabe von <RET> wird der alte Text beibehalten, durch Eingabe von Text dieser neu eingefügt.

Kundennummer
Eine Kundennummer kann ebenfalls zur Selektierung von Adressengruppen verwendet werden. Diese Nummer kann bis zu 6 Stellen umfassen und wird wie der vorher beschriebene Code behandelt. Eine Stelle kann mit (X) ausgeblendet werden.

Die Anpassung des Druckbildes an Aufklebe-Etiketten:

Der Zeilenvorschub zwischen zwei Adressen wird in Zeile 8390 ausgeführt. Durch Ändern der FOR..NEXT Schleife kann das Druckbild an die verschiedenen Drucker und Aufkleber Typen angepasst werden.

```
100  :
101  :
110  REM  CONTROL LOGIC
120  GOSUB 600: REM  INIT
200  GOSUB 1000: REM   MENU
300  ON I GOSUB 4000,5000,6000,2000,3000,
     2500,2000,8700,9000,3100
310  IF J >  = 0 THEN 200
330  PRINT  CHR$ (13);"ZUM WEITERMACHEN:
     EINE TASTE DRUECKEN"
340  CALL  - 756
350  GOTO 200
360  :
361  :
600  REM  INIT
620  DIM I$(499,1): REM  INDEX TABLE
630  F$ = "": REM  FILE NAME
635  C$ = "                                ":
     REM  BLANKS
637  E$ = "": REM  0 BLANKS
640  D$ =  CHR$ (4): REM  CTRL-D
650  N = 0: REM  NEXT ENTRY IN INDEX
660  PRINT D$;"NOMON C,I,O"
670  RETURN
680  :
681  :
1000 REM  MENU
1100 B$ = "IAORLQEDCT"
1110 CALL  - 936
1120 PRINT "ANGABE DER DATEI UND"
1130 PRINT "EINTRAGS-AENDERUNGEN"
1140 VTAB 4
1150 PRINT "   DATEIVERAENDERUNGEN:
     "
1155 PRINT
1160 PRINT "E  EROEFFNEN EINER NEUEN DATEI"
1170 PRINT "R  DATEI EINLESEN"
1180 PRINT "L  LISTE ALLER EINTRAEGE"
1182 PRINT "T  TEIL-LISTEN "
1185 PRINT "D  DRUCKEN VON EINTRAEGEN"
1190 PRINT
1200 PRINT "   EINTRAGS-AENDERUNGEN"
1205 PRINT
```

```
1210  PRINT "I  EINFUEGEN EINES NEUEN EINTRAGS"
1220  PRINT "A  AUSGABE..."
1230  PRINT "O  LOESCHEN... EINES EINTRAGS"
1235  PRINT "C  AENDERN..."
1240  PRINT
1250  PRINT "Q  BEENDEN DES PROGRAMMS"
1260  GOSUB 3990
1280  INPUT "";A$
1290  FOR I = 1 TO 10: IF  LEFT$ (A$,1)
        =  MID$ (B$,I,1) THEN 1320
1300  NEXT I
1310  PRINT "";: GOTO 1260
1320  CALL  - 936
1350  RETURN
2000  REM  START A NEW FILE
2010  REM  OR GET AN OLD FILE
2015  REM  FIRST CLOSE CURRENT FILE
2030  IF F$ <  > "" THEN  GOSUB 11500
2050  PRINT "EINGABE DES DATEINAMENS:
        ";
2060  INPUT "";F$: REM  NAME OF NEW FILE
2070  GOSUB 11000: REM  OPEN NEW FILE
2080  REM
2090  GOTO 310
2095  :
2096  :
2500  REM  QUIT
2510  REM  CLOSE FILE
2520  IF F$ <  > "" THEN  GOSUB 11500
2530 END
2540  :
2541  :
3000  REM  LIST A RECORD UNFORMATTED
3005  XX =  - 1
3010  IF N < 1 THEN  PRINT "KEINE EINTRAEGE
      VORHANDEN": GOTO 3090
3020  GOSUB 8500
3022  IF XX < 0 THEN  GOTO 3026
3024  CALL  - 936: PRINT D$;"PR#1"
3026  PRINT "DATEINAME: ";F$; CHR$ (13)
3030  FOR J = 0 TO N - 1
3050    IF  PEEK ( - 16384) > 127 THEN
        POKE  - 16368,0:
        CALL  - 756
```

```
3060    :: GOSUB 19100
3070    :: PRINT  LEFT$ (A$,41); CHR$ (13);:
        REM     LISTE DER EINT            RAEGE
3080  NEXT J
3090  J =  - 1
3094  GOSUB 8600
3095  RETURN
3096  :
3097  :
3100  REM  TEIL-LISTEN
3110  VTAB 4: PRINT "    AUSWAHL NACH":
      PRINT
3120  PRINT "K   KUNDENNUMMERN"
3130  PRINT "C   CODE"
3150  GOSUB 3990
3160  INPUT "";A$
3170  IF A$ = "C" THEN CC = 1:
      GOSUB 8000: GOTO 3199
3180  IF A$ = "K" THEN KK = 1:
      GOSUB 3300: GOTO 3199
3199  RETURN
3200  IF XX < 0 THEN  GOTO 3230
3210  CALL  - 936: PRINT D$;"PR#1"
3220  XX =  - 1
3230  PRINT  LEFT$ (A$,41); CHR$ (13);
3240  RETURN
3300  REM  AUSWAHL NACH KUNDENNUMMERN
3310  VTAB 4: PRINT "    AUSWAHL NACH KUNDENNUMMERN":
      PRINT
3320  PRINT "(X)  FUER NICHTBEACHTUNG DIESER"
3330  PRINT "     DIESER STELLE":
      PRINT
3340  INPUT "KUNDENNUMMER:
      ";CO$: IF CO$ = "" OR CO$ = "X" THEN
      PRINT "": GOTO 3340
3350  CO$ =  LEFT$ (CO$ + C$,6):
      XX =  - 1
3355  GOSUB 8500
3360  FOR J = 0 TO N - 1
3370    GOSUB 19100
3380    B$ =  MID$ (A$,31,6):
        P1 = 0
3390    FOR P = 1 TO 6
```

```
3400      IF  MID$ (CO$,P,1) = "X" THEN
          GOTO 3420
3410      IF  MID$ (CO$,P,1) < > MID$ (B$,
          P,1) THEN P1 = 1
3420    NEXT P
3430    IF P1 = 1 THEN  GOTO 3450
3435    IF KK = 1 THEN  GOSUB 3200:
        GOTO 3450
3440    GOSUB 8200
3450  NEXT J
3460  GOSUB 8600
3470  J =  - 1:KK =  - 1:
      RETURN
3500  REM  DRUCKEN EINES EINTRAGS
3510  CALL  - 936: GOSUB 7000:
      GOSUB 19000
3520  XX =  - 1:B$ = A$:
      GOSUB 8500
3525  CALL  - 936
3530  GOSUB 8200:J =  - 1:
      GOSUB 8600: RETURN
3990  VTAB 22: HTAB 1: CALL  - 958
3995  PRINT "EINGABE DES BUCHSTABENS UND
      <RET>";
3999  RETURN
4000  REM  ADD RECORD
4010  GOSUB 7000
4020  A$ = V$:
4022  INPUT "KUNDENNUMMER:
        ";B$
4024  A$ = A$ +  LEFT$ (B$ + C$,6)
4030  INPUT "CODE:          ";B$
4040  A$ = A$ +  LEFT$ (B$ + C$,5)
4050  INPUT "TITEL          ";B$
4060  A$ = A$ +  LEFT$ (B$ + C$,15)
4070  INPUT "FIRMA          ";B$
4080  A$ = A$ +  LEFT$ (B$ + C$,25)
4090  INPUT "STRASSE        ";B$
4100  A$ = A$ +  LEFT$ (B$ + C$,25)
4110  INPUT "STADT          ";B$
4120  A$ = A$ +  LEFT$ (B$ + C$,25)
4130  INPUT "TELEFON        ";B$
4140  A$ = A$ +  LEFT$ (B$ + C$,15)
```

```
4150  GOSUB 16000: REM  ADD RECORD
4160  PRINT : INPUT "WEITERE EINTRAEGE(J/N)
      ";X$
4170  IF X$ = "J" THEN  CALL  - 936:
      GOTO 4000
4180  RETURN
4190  :
4200  :
5000  REM  ************************
5001  REM        AUSGABE
5002  REM  ************************
5010  PRINT "A  AUSGABE ALLER EINTRAEGE"
5020  PRINT "E  AUSGABE EINES EINTRAGS"
5030  GOSUB 3990: INPUT "";A$
5040  GOSUB 8500: CALL  - 936
5050  IF A$ = "A" THEN  GOSUB 5200:
      GOTO 5190
5060  IF A$ = "E" THEN  GOSUB 5300:
      GOTO 5190
5070  PRINT "": GOTO 5030
5190  IF XX = 1 THEN  GOSUB 8600
5199  RETURN
5200  IF N < 1 THEN  PRINT "*** KEINE EINTRAEGE":
      RETURN
5210  IF XX = 1 THEN  PRINT D$;"PR#1"
5220  FOR J = 0 TO N - 1
5230    GOSUB 19100: GOSUB 5550:
      NEXT J
5240  RETURN
5300  GOSUB 5500
5310  IF XX = 1 THEN  PRINT D$;"PR#0"
5320  INPUT "WEITERE EINTRAEGE (J/N)";X$
5330  IF X$ <  > "J" THEN  RETURN
5340  CALL  - 936: GOTO 5300
5500  REM  DISPLAY RECORD
5510  JJ = 0:JA = 0
5520  GOSUB 7000
5530  GOSUB 19000
5540  IF J < 0 THEN JA =  - 1:
      GOTO 5999
5545  IF XX = 1 THEN  PRINT D$;"PR#1"
5550  B$ = A$:Z$ = I$(J,1):
      JJ = J
```

```
5560  PRINT  CHR$ (13);E$ +  LEFT$ (B$,30);
      CHR$ (13);: REM  NAME
5570  Rl = 31
5580  PRINT E$;"KUNDENNUMMER:
      "; MID$ (B$,Rl,6); CHR$ (13);:
      Rl = Rl + 6
5590  PRINT E$;"CODE: " +  MID$ (B$,Rl,5);
      CHR$ (13);:Rl = Rl + 5
5600  PRINT E$ +  MID$ (B$,Rl,15); CHR$ (13);:
      Rl = Rl + 15: REM  TITEL
5610  PRINT E$ +  MID$ (B$,Rl,25); CHR$ (13);:
      Rl = Rl + 25: REM  FIRMA
5620  PRINT E$ +  MID$ (B$,Rl,25); CHR$ (13);:
      Rl = Rl + 25: REM  STRASSE
5630  PRINT  MID$ (B$,Rl,25); CHR$ (13):
      Rl = Rl + 25: REM  STADT
5640  PRINT E$ + "TEL:" +  MID$ (B$,Rl,15)
5999  RETURN
6000  REM  ERASE RECORD
6010  GOSUB 7000
6020  GOSUB 18000
6030  RETURN
7000  REM  GET KEY
7010  VTAB 6
7020  PRINT "EINGABE:"
7030  INPUT "FAMILIENNAME   ";B$
7040  V$ =  LEFT$ (B$ + C$,20)
7050  INPUT "VORNAME        ";B$
7060  V$ = V$ +  LEFT$ (B$ + C$,10)
7070  RETURN
7080  :
7081  :
8000  REM  DRUCKEN
8005  XX =  - 1:C = 1: CALL  - 936
8010  PRINT "CODE EINGABE:
      <RET> FUER ALLE ADRESSEN"
8020  PRINT "             :
      <X> FUER BELIEBEIGEN"
8030  PRINT "              CODE-PARAMETER"
8040  INPUT "CODE=";CO$:
      IF CO$ = "" THEN C = 0
8041  GOSUB 8500: CALL  - 936
8043  IF C = 0 THEN  GOTO 8050
```

```
8045  C = 1:CO$ =  LEFT$ (CO$ + C$,5)
8050  IF N < 1 THEN  PRINT "KEINE EINTRAEGE":
      GOTO 8180
8060  FOR J = 0 TO N - 1
8064    GOSUB 19100
8070    IF C = 0 THEN  GOTO 8160
8100    B$ =  MID$ (A$,37,5):
        P1 = 0
8110    FOR P = 1 TO 5
8120      IF  MID$ (CO$,P,1) = "X" THEN
          GOTO 8140
8130      IF  MID$ (CO$,P,1) <  >  MID$ (B$,
          P,1) THEN P1 = 1
8140    NEXT P
8150    IF P1 = 1 THEN  GOTO 8170
8155    IF CC > 0 THEN  GOSUB 3200:
        GOTO 8170
8160    GOSUB 8200
8170  NEXT J
8172  GOSUB 8600
8175  J =  - 1
8177  CC =  - 1
8180  RETURN
8200  REM  AUSGABE
8202  IF XX < 0 THEN  GOTO 8210
8204  CALL  - 936: PRINT D$;"PR#1"
8208  XX =  - 1:R3 = 31
8210  R3 = 31:B$ =  MID$ (A$,R3 + 6,1)
8220  IF B$ = "M" THEN  PRINT E$ + "HERRN
      ";
8222  IF B$ = "W" THEN  PRINT E$ + "FRAU
      ";
8224  IF B$ = "F" THEN  PRINT E$ + "FIRMA";
8226  IF B$ = "O" THEN  PRINT  CHR$ (13);:
      GOTO 8270
8250  B$ =  MID$ (A$,R3 + 11,15)
8260  IF B$ <  >  LEFT$ (C$,15) THEN  PRINT B$;
8270  PRINT  CHR$ (13);
8272  AA$ =  MID$ (A$,21,10)
8274  FOR II = 1 TO 10
8276    IF  MID$ (AA$,II,1) = " " THEN
        IF  MID$ (AA$,II + 1,1) = "
        " THEN  GOTO 8280
```

```
8278  NEXT II
8280  PRINT E$; MID$ (A$,21,II); LEFT$ (A$,
      20); CHR$ (13);
8290  B$ =  MID$ (A$,R3 + 26,25)
8300  IF B$ <  > C$ THEN  PRINT E$;B$;
8310  PRINT  CHR$ (13);
8320  PRINT E$; MID$ (A$,R3 + 51,25); CHR$ (13);
8330  PRINT  CHR$ (13); MID$ (A$,R3 + 76,
      25);
8390  FOR G = 1 TO 2: PRINT  CHR$ (13);:
      NEXT
8399  RETURN
8500  REM  DRUCKER EIN
8510  PRINT : INPUT "DRUCKER EIN (J/N) ";X$
8520  IF X$ <  > "J" THEN  GOTO 8599
8525  PRINT " DRUCKER ";:
      INVERSE : PRINT "EIN":
      NORMAL
8530  XX = 1
8599  RETURN
8600  REM  DRUCKER AUS
8610  PRINT D$;" PR#0":XX =  - 1:

8620  PRINT "DRUCKER AUS"
8630  RETURN
8700  GOSUB 9700
8710  ON I GOSUB 3500,8000,3300
8820  RETURN
9000  REM  AENDERN EINES EINTRAGES
9010  GOSUB 5500: REM  ANZEIGEN
9015  IF JA < 0 THEN  GOTO 9160
9018  PRINT : PRINT : PRINT
9019  R3 = 31
9020  INPUT "KUNDENNUMMER:
      ";B$
9025  IF B$ <  > "" THEN R1 = R3:
      R2 = 6: GOSUB 9500
9030  INPUT "CODE   :";B$
9040  IF B$ <  > "" THEN R1 = R3 + 6:
      R2 = 5: GOSUB 9500
9050  INPUT "TITEL  :";B$
9060  IF B$ <  > "" THEN R1 = R3 + 11:
      R2 = 15: GOSUB 9500
```

```
 9070  INPUT "FIRMA  :";B$
 9080  IF B$ < > "" THEN R1 = R3 + 26:
       R2 = 25: GOSUB 9500
 9090  INPUT "STRASSE:";B$
 9100  IF B$ < > "" THEN R1 = R3 + 51:
       R2 = 25: GOSUB 9500
 9110  INPUT "STADT  :";B$
 9120  IF B$ < > "" THEN R1 = R3 + 76:
       R2 = 25: GOSUB 9500
 9130  INPUT "TELEFON:";B$
 9140  IF B$ < > "" THEN R1 = R3 + 101:
       R2 = 15: GOSUB 9500
 9150  J = JJ:I$(J,1) = Z$:
       GOSUB 19200
 9160  J =  - 1:JJ = 0:N1 = 1
 9164  INPUT "WEITERE AENDERUNGEN (J/N)";JA$
 9166  IF  LEFT$ (JA$,1) = "J" THEN  CALL
       - 936: GOTO 9000
 9170  RETURN
 9500  Z1$ =  LEFT$ (A$,R1 - 1)
 9510  Z2$ =  LEFT$ (B$ + C$,R2)
 9520  Z3$ =  RIGHT$ (A$,R2)
 9530  A$ = Z1$ + Z2$ + Z3$
 9540  RETURN
 9700  B$ = "ECK"
 9710  VTAB 4: PRINT "    DRUCKEN VON":
       PRINT
 9720  PRINT "E   EINZELNEN EINTRAEGEN"
 9730  PRINT "C   AUSWAHL NACH CODE"
 9740  PRINT "K   AUSWAHL NACH KUNDENNUMMERN"
 9750  VTAB 22: HTAB 1: CALL  - 958
 9760  PRINT "EINGABE DES BUCHSTABENS UND
       <RET>";
 9770  INPUT "";A$
 9780  FOR I = 1 TO 3: IF  LEFT$ (A$,1) =
         MID$ (B$,I,1) THEN 9800
 9790  NEXT I: PRINT "": GOTO 9760
 9800  CALL  - 936: RETURN
10000  REM  INDEX SEQUENTIAL ROUTINES
10120  REM  A$ RECORD DATA
10130  REM  D$ CTRL D
10140  REM  F$ FILE NAME
10150  REM  I$ INDEX TABLE
```

```
10151  :
10152  :
11000  REM  OPEN
11030  N1 = 0:N2 = 0:J = 0
11060  PRINT D$;"OPEN";F$;"INDEX"
11080  PRINT D$;"READ";F$;"INDEX"
11110  ONERR  GOTO 11140
11120  INPUT N,M,N2
11125  IF I = 7 THEN  PRINT "*** DATEINAME
       SCHON VORHANDEN":N = 0:
       J =  - 1: POKE 216,0:
       GOTO 2080
11130  GOTO 11150
11140  IF I = 4 THEN  PRINT "*** DATEI NICHT
       GEFUNDEN ":J =  - 1:
       POKE 216,0: GOTO 2080
11142  M = 148
11145  N1 = 1
11150  POKE 216,0
11160  J1 = 0
11170  IF J1 >  = N THEN 11210
11180  :: INPUT I$(J1,0),I$(J1,1)
11190  ::J1 = J1 + 1
11200  :: GOTO 11170
11210  REM  CLOSE INDEX FILE
11220  PRINT D$;"CLOSE";F$;"INDEX"
11223  IF J < 0 THEN 11290
11225  REM  OPEN DATA FILE
11230  PRINT D$;"OPEN";F$;",L";M + 1
11290  GOTO 2080
11300  :
11301  :
11500  REM  CLOSE
11510  GOSUB 12000
11520  PRINT D$;"CLOSE";F$
11530  RETURN
11540  :
11541  :
12000  REM  WRITE INDEX
12020  IF N1 = 0 THEN 12120
12030  PRINT D$;"OPEN";F$;"INDEX"
12040  PRINT D$;"WRITE";F$;"INDEX"
12050  PRINT N;",";M;",";N2
```

```
12060  J1 = 0
12070  IF J1 >  = N THEN 12110
12080  :: PRINT I$(J1,0);",";I$(J1,1)
12090  ::J1 = J1 + 1: GOTO 12070
12110  PRINT D$;"CLOSE";F$;"INDEX"
12120  RETURN
12300  REM  FIND ENTRY
13000  REM  ENTRY
13060  J = 0
13080  IF N <  = 0 THEN 13200
13100  IF V$ <  = I$(0,0) THEN 13200
13120  J1 = 0:J2 = N
13130  ::J =  INT ((J1 + J2) / 2)
13140  :: IF V$ = I$(J,0) THEN 13200
13150  :: IF V$ < I$(J,0) THEN J2 = J:
       GOTO 13180
13160  ::J1 = J
13180  :: IF J <  >  INT ((J1 + J2) / 2)
       THEN 13130
13190  J = J + 1
13200  RETURN
14000  REM  EXPAND INDEX
14100  IF N > 499 THEN  PRINT "";"*** FEHLER
       INDEXTABELLE VOLL":J =  - 1:
       GOTO 14150
14105  J1 = N
14110  IF J1 < J + 1 THEN 14150
14120  ::I$(J1,0) = I$(J1 - 1,0)
14130  ::I$(J1,1) = I$(J1 - 1,1)
14140  ::J1 = J1 - 1: GOTO 14110
14150  RETURN
15000  REM  COMPACT INDEX TABLE
15060  N = N - 1
15070  IF J >  = N THEN 15120
15080  FOR J1 = J TO N
15090    ::I$(J1,0) = I$(J1 + 1,0)
15100    ::I$(J1,1) = I$(J1 + 1,1)
15110  NEXT J1
15120  RETURN
16000  REM  WRITE DIREKT
16110  GOSUB 13000
16130  IF V$ = I$(J,0) THEN  PRINT "***
       GLEICHER NAME SCHON VORHANDEN":
       J =  - 1: GOTO 16300
```

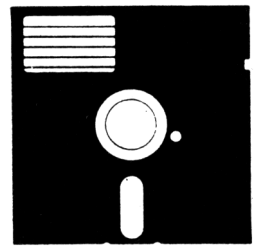
```
16160  GOSUB 14000
16170  IF J < 0 THEN 16300
16190  I$(J,0) = V$
16200  I$(J,1) =  STR$ (N)
16205  N2 = J
16210  GOSUB 19200:
16220  N = N + 1
16230  N1 = 1
16300  RETURN
18000  REM  DELETE
18110  GOSUB 13000
18130  IF V$ <  > I$(J,0) THEN  PRINT "***
       EINTRAG NICHT GEFUNDEN ***:
       J=-1:GOTO 18440
18150  N1 = 1
18210  IF N2 = J THEN 18330
18275  J1 = J:J = N2
18280  GOSUB 19100:J = J1
18300  GOSUB 19200
18305  I$(N2,1) = I$(J,1)
18310  GOSUB 15000
18320  GOTO 18430
18330  GOSUB 15000
18340  A$ =  STR$ (N - 1)
18350  FOR J = 0 TO N - 1
18360    :: IF I$(J,1) = A$ THEN 18430
18370  NEXT J
18390  PRINT "*** FEHLER IN NAMENSLISTE
       ***"
18400  PRINT "EINTRAEGE KOENNEN FALSCH GEORDNET
       SEIN"
18405  N2 = N - 1
18410  J =  - 1: GOTO 18440
18430  N2 = J
18440  RETURN
19000  REM  READ
19010  GOSUB 13000
19020  IF V$ <  > I$(J,0) THEN  PRINT "***
       EINTRAG NICHT GEFUNDEN":
       J =  - 1: GOTO 19050
19040  GOSUB 19100
19050  RETURN
19100  REM  READ
```

```
19120  PRINT D$;"READ";F$;",R";I$(J,1)
19130  INPUT A$: PRINT D$
19150  RETURN
19200  REM  WRITE
19220  PRINT D$;"WRITE";F$;",R";I$(J,1)
19240  PRINT  LEFT$ (A$,M):
       PRINT D$
19250  RETURN
19260 END
20000  D$ =  CHR$ (4)
20010  PRINT D$;"OPEN  TEST"
20020  PRINT D$;"READ TEST"
20030  INPUT A$
20040  PRINT D$;"CLOSE  TEST"
20050  PRINT A$
20060 END
```

Abbildung 52: ADREKART.

Notizen

Alle Programme aus diesem Buche auf Diskette



Für denjenigen der nicht die Zeit und Muse hat, die Programme einzugeben, halten wir eine Diskette bereit, die alle Programme aus diesem Buche enthält.

Wir senden Ihnen diese gerne gegen Vorauszahlung von DM 149,– incl. Versand und Verpackung zu.

Verwenden Sie dieses Blatt als Bestellschein.

. . . . . .
Name

. . . . . .
Vorname

. . . . . .
Straße

. . . . . .
PLZ Ort

☐ Ich bestelle (Best.-Nr. 2241) die Diskette, voll mit Programmen zum Buch # 224 zu DM 149, – incl. Porto und Verpackung. Den Betrag von DM 149, – habe ich heute auf Ihr Postscheck-Konto München 15994–807 überwiesen.

☐ Bitte liefern Sie per Nachnahme. Hier kommen noch NN-Gebühren in Höhe von 6,50 DM hinzu.

. . . . . .
Datum Unterschrift (f. Jugendliche unter 18 Jahre der Erziehungsberechtigte

Notizen

HOFACKER Bücher
für den
professionellen Anwender

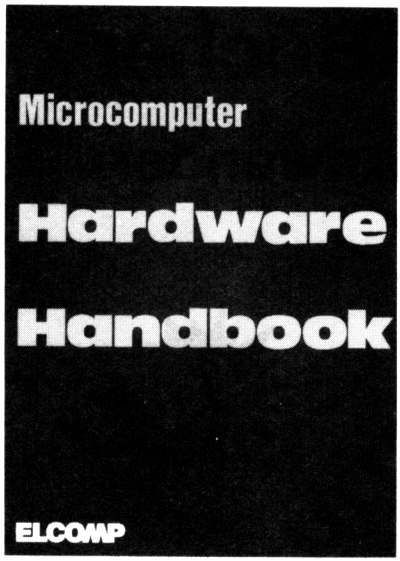
Endlich ein
umfangreiches deutsches
Buch über Funktionsanalyse

Das Fachbuch für den COMPUTER ANWENDER

Funktionsanalyse
Grundlagen und Anwendung

Dies ist eines der umfangreichsten, deutschen Werke über das Verständnis von Methoden und Verfahren beim Testen integrierter Schaltungen und Software. Der Begriff "digitale Schaltung" erstreckt sich hier vom einfachen Schaltnetz bis hin zu komplexen Mehrrechnersystemen auf Mikrocomputerebene. Aus dem Inhalt: Testverfahren und Funktion zugehöriger Geräte, Signaturanalyse, Datenkompression, Logik Analyse, Prüfmustererzeugung, In Circuit Emulation, Entwicklungssysteme, rechnergesteuerter Messplatz, rechnergestützte Diagnose digitaler Schaltungen, Softwarehilfen u. Softwaretest. Dieses Buch gehört auf den Arbeitsplatz jedes Hardwareentwicklers und Laboringenieurs. Es geht u. a. auf die 8088/8086/8089 Prozessoren ein.
Verfasser: Josef Koller, 287 Seiten

Best.-Nr. 136 **79,– DM**



Microcomputer Hardware Handbook

Eine sehr umfangreiche (840 Seiten) Sammlung der Daten und Spezifikationen der wichtigsten Microprozessor Bausteine.. Auch TTL, CMOS, LPS, FAST, Linear und Peripheriebausteine sind enthalten. Dieses Buch braucht jed. PC Service Mann und Hardware Freak. Komplette Datenblätter vom 6502, Z80, 6800, 6809, 8088, 8086, 8085, u. v. a. Auch zugehörige periphere Schaltungen u. Supportschaltungen sind enthalten. EPROMs, RAMs, ROMs.

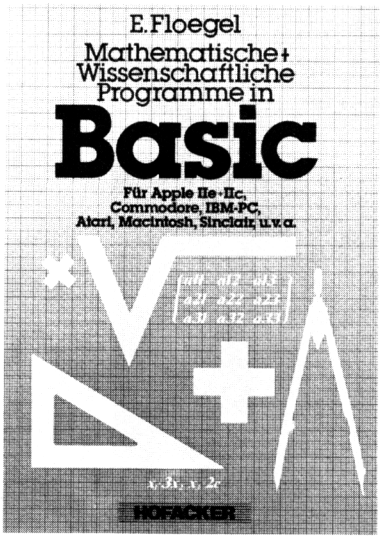
Best.-Nr. 29 **49, – DM**

BASIC für Fortgeschrittene

Die Hohe Schule für BASIC. Jeder der seinen PC besser nutzen will braucht dieses Buch. Commodore BASIC, MBASIC, Macintosh BASIC, TANDY BASIC, IBM PC BASIC viele Programmbeispiele. Künstliche Intelligenz-Anwendungsbeispiele, Grundlagen Floppy Disk Technik, Dateien auf Diskette und Cassette, in Commodore BASIC, CP/M und MS-DOS, ATARI und APPLE. Strategisches Programmieren, der binäre Baum und seine Anwendung, Sortierverfahren. Das sind nur einige Themen aus diesem sehr umfangreichen u. breitgestreuten Anwendungsfeld. Dieses Buch ist ein Muss für jeden BASIC Programmierer, gleich welchen Computer er besitzt oder noch kaufen will. (226 S.)

Best.-Nr. 122 **39, – DM**

NEUHEITEN:



Mathematische und wissenschaftliche Programme in BASIC

Verfasser: Dipl. Ing. E. Flögel (141 Seiten)

Das ist das Buch für jeden Schüler, Studenten und alle, die sich für Mathematik und deren Anwendung auf dem Personalcomputer interessieren. Die Programme sind in Standard BASIC verfasst, so daß sie auf allen Personalcomputern mit Microsoft BASIC arbeiten. Ideal für Apple IIe und IIc, sowie für IBM PC und Commodore.

Aus dem Inhalt: Elementare Mathematik, Funktionen und Polynome, komplexe Zahlen, Vektoren, Matrizen, lineare Gleichungssysteme, Ausgleichsrechnung, numerische Integration von Differentialgleichungen, Lösung des Dirchiletproblems durch Differenzenverfahren. Lineare Regression, lineare Planungsrechnung, Algebra.

Best.-Nr. 102 **29,80 DM**

Alle Programme aus Buch Nr. 102 auf Diskette:

Best.-Nr. 1021 für Apple IIe und IIc **99,– DM**
Best.-Nr. 1024 für IBM PC u. 100% komp. **99,– DM**

Kombinationspakete bestehend aus Buch und Diskette. Verpackt im praktischen und formschönen Kunststoff-Ordner:

Best.-Nr. 1023 für APPLE IIe und IIc **119,– DM**
Best.-Nr. 1022 für IBM und 100% komp. **119,– DM**

E. FLOEGEL

Statistik in BASIC
Einführung und praktische Anwendungen

Verfasser: Dipl. Ing. E. Flögel (209 Seiten)

Hier finden Sie das Statistik-Buch, auf das Sie schon lange gewartet haben. Es werden nicht nur Programme und deren Ergebnisse mitgeteilt, sondern auch das ganze Warum und Drumherum mit beleuchtet. Die Programme sind vorwiegend in Microsoft BASIC geschrieben, so daß sie auf nahezu allen Personalcomputern arbeiten müßten. Bei Grafikprogrammen wurden Versionen für den IBM PC und den APPLE IIe oder IIc mit angegeben.

Ganz besonders ausführlich wurde der Teil Grafik behandelt. Aus dem Inhalt: Wahl der Daten, graphische Darstellung, Berechnung statistischer Kenngrößen, Wahrscheinlichkeitsrechnung, diskrete, statistische Verteilungen, statistische Testverfahren, parameterfreie Testverfahren, lineare Regression, mehrfache, lineare Regression, nichtlineare Regression, Autokorrelation, Trendanalyse und am Schluß ein ganzes Kapitel über Statistik und Glücksspiel.

Von diesem Buch werden Sie begeistert sein.

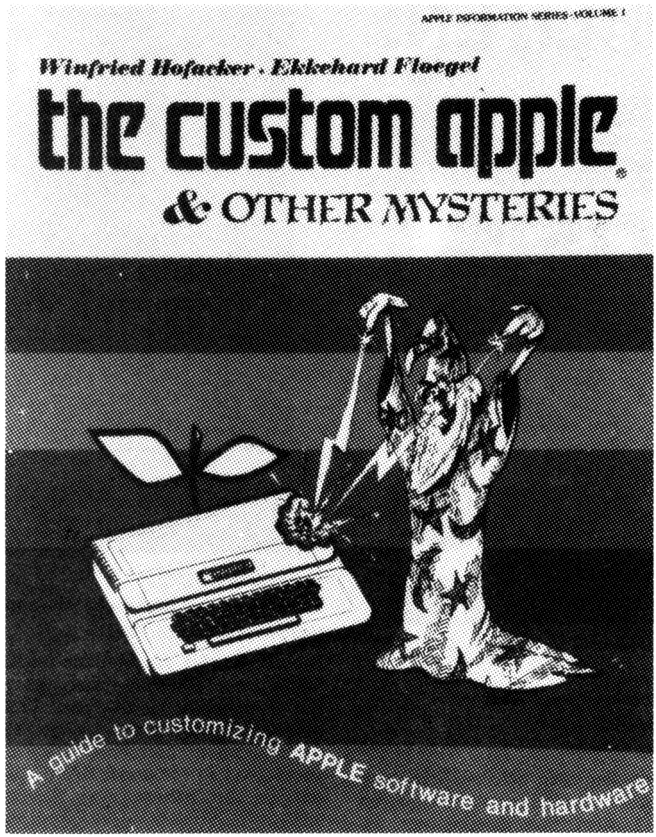
Best.-Nr. 188 **39,– DM**

Die Programme aus diesem Buch haben wir für Sie auf Disketten bereitgestellt.

Best.-Nr. 1882 Diskette für IBM PC **79,– DM**
Best.-Nr. 1883 Diskette für APPLE IIe u. c **79,– DM**

Kombinationspaket bestehend aus Buch Nr. 188 und Diskette:

Best.-Nr. 682 für Apple IIe und IIc **99,– DM**
Best.-Nr. 1881 für IBM PC **99,– DM**



The Custom APPLE & other Mysteries

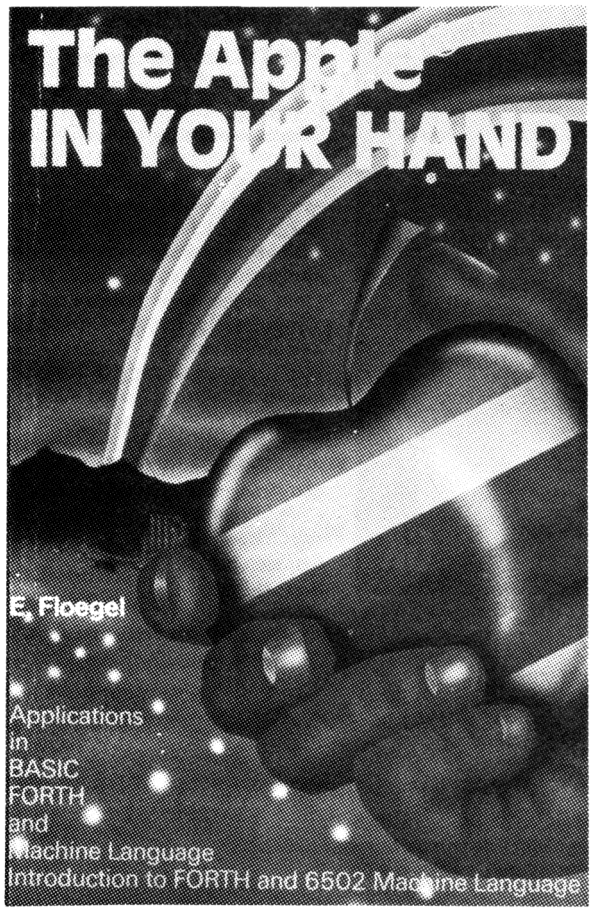
Verfasser: W. Hofacker, E. Flögel englisch 190 Seiten

Dieses Buch bringt auf 190 Seiten Großformat eine Vielzahl von Erweiterungsprojekten für Ihren APPLE II.

6522 Ein-/Ausgabekarte, Tonerzeugung mit AY-3-8912, Analog/Digital-Wandler, EPROM-Burner, Anschluß des 8253 von Intel an den APPLE II, Schrittmotorsteuerungen mit Apple II in BASIC, PASCAL und FORTH.

Zu den einzelnen Projekten sind die notwendigen Platinenvorlagen sowie die Software im Buch enthalten. Alle Platinen zum Buch können vom HOFACKER Verlag sofort ab Lager bezogen werden.

Best.-Nr. 680 **79,– DM**



The Apple in Your Hand
Applications in BASIC, FORTH and Machine Language, Introduction to FORTH and 6502 Machine Language

Verfasser: E. Flögel englisch 220 Seiten

Dieses Buch ist ein Muß für jeden Apple Computer-Besitzer. Eine Einführung in 6502 Maschinensprache und FORTH machen dieses Werk zu einem wertvollen Arbeitsbuch und Nachschlagewerk. Die vielen wertvollen Anwenderprogramme sorgen für den Vorrat, den der Apple-Anwender immer dann braucht, wenn es darum geht, nicht routinemäßige Codieraufgaben sinnvoll zu lösen.

Best.-Nr. 178 **39,– DM**

FORTH Anwendungsbeispiele

von E. Flögel

Datenverwaltungs-, Geschäftsprogramme, Programmiertechniken, Unterhaltungsprogramme, Künstliche Intelligenz.

Dies ist das erste Buch (206 Seiten) weltweit, welches Anwendersoftware in der Programmiersprache FORTH enthält.

FORTH hat sich in den letzten Jahren immer mehr Freunde geschaffen. In erster Linie wegen seiner hohen Portabilität. Aus dem Inhalt: Ein- u. Ausgabe von Daten, Zeichenketten in FORTH, verkettete Listen, ein kompl. Macroassembler, ein kleines Spiel, usw. sowie auch ein kompl. Geschäftsprogr., bestehend aus Adressen-, Lagerverwaltung und Fakturierung. Dieses Paket dürfte eine echte Sensation sein, da der Anwender hier den kompl. Quelltext eines Programmes erhält,, dessen Entwicklungskosten eine sechsstellige Zahl betragen haben.

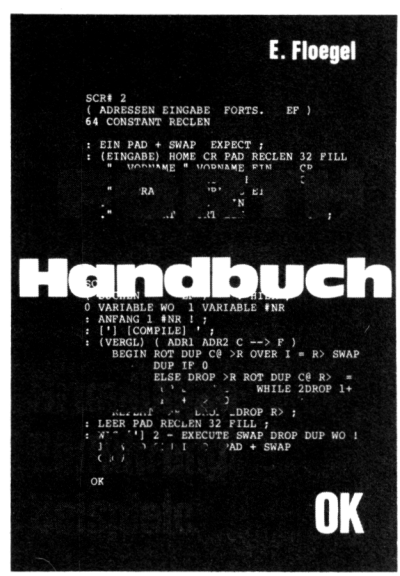
Best.-Nr. 200 **49,– DM**

FORTH Handbuch

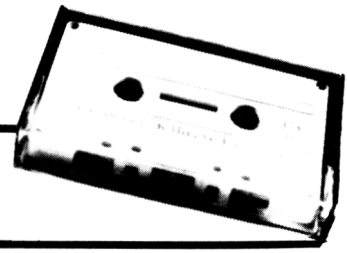
von E. Flögel

Dies ist das führende deutschsprachige Werk über diese sehr leistungsfähige Programmiersprache. Nach einer sehr guten Einführung bringt der Autor Beispiele aus der Praxis. Adressenverwaltung in FORTH mit virtueller Speicherverwaltung u. v. a. mehr. Ein Muß für jeden, der FORTH kennenlernen will. FORTH ist eine extrem leistungsfähige Programmiersprache mit Interpreter- u. Compilereigenschaften. Die Codierung ist sehr effektiv u. Programme lassen sich in einem Bruchteil der Zeit erstellen, die man in jeder anderen, höheren Programmiersprache benötigen würde. FORTH-Programme lassen sich auf fast allen Rechnern, trotz unterschiedlicher CPU, nahezu ohne Änderungen implementieren. Die Sprache, die auch Sie unbedingt erlernen sollten. Dieses Buch (172 Seiten) ist Ihre Fahrkarte für den Einstieg in diese leistungsfähige Sprache.

Best.-Nr. 137 **49,– DM**



Leercassetten für Microcomputer



C-10

Die ideale Cassettenlänge für Ihren Personalcomputer!

Praktisch – handlich und betriebssicher.

Kassetten mit nur 10 Minuten Spieldauer (2 x 5 Minuten) haben sich zur Aufzeichnung von Daten im Microcomputerbereich bestens bewährt.

Vorteile der C-10 Computer Cassette vom HOFACKER Verlag:

- ★ weniger Bandsalat
- ★ kurze Rückspulzeiten
- ★ schnelles Auffinden von Programmen
- ★ bessere Gleichlaufeigenschaften
- ★ einfache Programmverwaltung
- ★ extrem hoch aussteuerbares Bandmaterial (AGFA)

Die C-10 HOFACKER Datencassette wird seit 1978 speziell für Microcomputeranwender produziert. Die Cassetten bieten ein Höchstmaß an Betriebssicherheit bezüglich fehlerfreier Aufnahme und Wiedergabe.

Diese Cassette eignet sich für alle heute am Markt befindlichen Personalcomputer, wie z. B.:

Commodore 64	**SHARP**	**TRS-80**
VC-20	**APPLE II**	**GENIE**
ATARI 600/800XL	**SINCLAIR**	**TI 99/4A**
ATARI 400/800	**IBM PC**	**OSBORNE**

Noch heute bestellen:	Best.-Nr. 8089 –	**1 Cassette**	**2,50 DM**
	Best.-Nr. 8100 –	**10 Cassetten**	**19,80 DM**
	Best.-Nr. 8096 –	**100 Cassetten**	**149,00 DM**

HOFACKER

Telefon
(0 80 24) 73 31
Telex
526973

Ing. W. Hofacker GmbH · Tegernseer Str. 18 · D–8150 Holzkirchen

Weitere interessante Bücher von Hofacker:

Best.-Nr.	Titel	Preis/DM
	BÜCHER in deutscher Sprache	
1	Transistor Berechnungs- und Bauanleitungsbuch-1	29,80
2	Transistor Berechnungs- und Bauanleitungsbuch-2	19,80
3	Elektronik im Auto	9,80
4	IC-Handbuch, TTL, CMOS, Linear	19,80
5	IC-Datenbuch, TTL, CMOS, Linear	9,80
6	IC-Schaltungen, TTL, CMOS, Linear	19,80
7	Elektronik Schaltungen	19,80
8	IC-Bauanleitungs-Handbuch	19,80
9	Feldeffekttransistoren	9,80
10	Elektronik und Radio	19,80
12	Beispiele integrierter Schaltungen (BIS)	19,80
13	HEH, Hobby Elektronik Handbuch	9,80
16	CMOS Teil 1, Einführung, Entwurf, Schaltbeispiele	19,80
17	CMOS Teil 2, Entwurf und Schaltbeispiele	19,80
18	CMOS Teil 3, Entwurf und Schaltbeispiele	19,80
19	IC-Experimentier Handbuch	19,80
20	Operationsverstärker	19,80
21	Digitaltechnik Grundkurs	19,80
22	Mikroprozessoren, Eigenschaften und Aufbau	19,80
23	Elektronik Grundkurs, Kurzlehrgang Elektronik	9,80
24	Programmieren in Maschinensprache mit Z80, II	29,80
25	68000 Microcomputer Einführung	39,00
26	Mikroprozessor, Teil 2	19,80
27	BASIC-M für 6800/09/68000 (Motorola)	29,80
28	Lexikon u. Wörterbuch f. Elektr. u. Mikroprozessor	29,80
30	Floppy Disk Selbstbau-Handbuch (i.V.)	49,00
31	57 Praktische Programme in BASIC	39,00
32	ATARI BASIC, für Selbststudium und Praxis	39,00
33	Microcomputer Programmierbeispiele	19,80
34	TINY-BASIC Handbuch	19,80
35	Der freundliche Computer	29,80
102	Mathematische + Wissenschaftliche Progr. i. BASIC	29,80
103	Oszillographen-Handbuch	19,80
104	Portable Computer Handbuch (i.V.)	39,00
108	Rund um den Spectrum (Progr., Tips und Tricks)	29,80
109	6502 Microcomputer Programmierung	29,80
111	Programmieren mit TRS-80 (GENIE)	29,80
112	PASCAL-Programmier-Handbuch	29,80
113	BASIC-Programmier-Handbuch (mit BASIC-Kurs)	19,80
114	Der Microcomputer im Kleinbetrieb	39,80
116	Einführung 16-Bit Microcomputer	29,80
118	Programmieren in Maschinensprache mit dem 6502	49,00
119	Programmieren in Maschinensprache mit Z80, I	39,00
120	Anwenderprogramme für TRS-80 und GENIE	29,80
121	Microsoft BASIC-Handbuch	29,80
122	BASIC für Fortgeschrittene	39,00
123	IEC-Bus Handbuch	19,80
124	Progr. in Maschinensprache mit Commodore-64	29,80
127	Einführung i.d. Microcomputer-Progr. mit 6800	49,00
128	Programmieren mit dem CBM	29,80
130	Programmierbeispiele für CBM	9,80
132	CP-M Handbuch	19,80
133	Handbuch für MS-DOS, IBM-PC	29,80
136	Funktionsanalyse	79,00
137	FORTH – Grundlagen, Einführung, Beispiele	49,00
139	BASIC für blutige Laien (speziell f. TRS-80, Genie)	19,80
140	Progr. i. BASIC u. Maschinencode mit dem ZX81	29,80
141	Progr. f. VC-20 (Spiele, Utilities, Erweiterungen	29,80
141	Progr. f. VC-20 (Spiele, Utilities, Erweiterungen)	29,80
143	35 Programme für den ZX81	29,80
144	33 Programme für den ZX-Spectrum	29,80
145	64 Programme für den Commodore-64	39,00
146	Hardware Erweiterungen für den Commodore-64	39,00
147	Beherrschen Sie Ihren Commodore-64	19,80
148	Programmierhandbuch für SHARP	49,00
149	Programme für TI 99/4A	49,00
175	Astrologie auf dem ATARI 800	49,00
187	Mehr als 29 Programme für den Commodore-64	29,80
188	Statistik in BASIC	39,00
189	Editor + Assembler, Macrofire	19,80
190	Das große Spielebuch für ATARI 600XL/800XL, I	29,80
200	FORTH-Anwendungsbeispiele	49,00
202	UNIX – Grundlagen und Anwendungen	39,00
204	Grafik und Ton mit dem Commodore-64	29,80
205	Das große Spielebuch für ATARI 600XL/800XL, II	29,80
210	Superprogramme für IBM-PC	29,80
212	Geschäftsprogramme für Commodore-64	39,00
213	Technische Gleichungssysteme in BASIC	49,00
217	Künstliche Intelligenz	19,80
224	Anwenderprogramme für Apple II c, und II e	19,80
8029	Z-80 Assembler-Handbuch	29,80
	BÜCHER in englischer Sprache	
29	Hardware Handbook	49,00
52	Small Business Programs for the IBM PC	29,80
151	8K Microsoft BASIC Reference Manual	9,80
152	Expansion Handbook for 6502 and 6800	19,80
156	Small Business Programs	29,80
158	The Second Book of Ohio Scientific	19,80
159	The Third Book of Ohio Scientific	29,80
160	The Fourth Book of Ohio Scientific	29,80
161	The Fifth Book of Ohio Scientific	19,80
162	ATARI Games in BASIC	19,80
164	ATARI-BASIC Learning by Using	19,80
166	Programming in 6502 Machinelanguage PET/CMB	49,00
169	How the Progr. y. ATARI i. 6502 Machinelanguage	29,80
170	FORTH on the ATARI – Learning by Using	29,80
171	See the Future with your ATARI! (Astrology)	49,00
172	Hackerbook I (Tricks + Tips for your ATARI)	29,80
173	PD-Program Descriptions (ATARI)	9,80
174	ZX-81/TIMEX Progr. i. BASIC a. Machine Lang.	4,90
176	Programs + Tricks for VIC's	29,80
177	CP/M – MBASIC and the OSBORNE	5,00
178	The APPLE in Your Hand	39,00
182	The Great Book of Games Vol. I – Games f. the C-64	29,80
183	More on the Sixtyfour (Commodore-64)	39,00
184	How to Progr. your C-64 i. 6502/10 Machinelang.	29,80
185	Commodore-64 Tune-up	39,00
186	Small Business Programs for the Commodore-64	29,80

Der HOFACKER-Verlag produziert und vertreibt neben einer sehr großen Auswahl an Fachbüchern für Elektronik und Microcomputertechnik noch:

- Leerplatine und Bauanleitungen für Zusatzeinrichtungen für Ihren Personalcomputer, sowie
- Programme (Software) und Leercassetten (C-10) für die bedeutenden Personalcomputer.

(i.V. bedeutet: Buch ist in Vorbereitung)

HOFACKER

HOLZKIRCHEN SINGAPORE LOS ANGELES

ISBN 3-88963-224-6